UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3

Guía para el aseguramiento de la seguridad lógica en el desarrollo de sistemas informáticos de gobierno electrónico

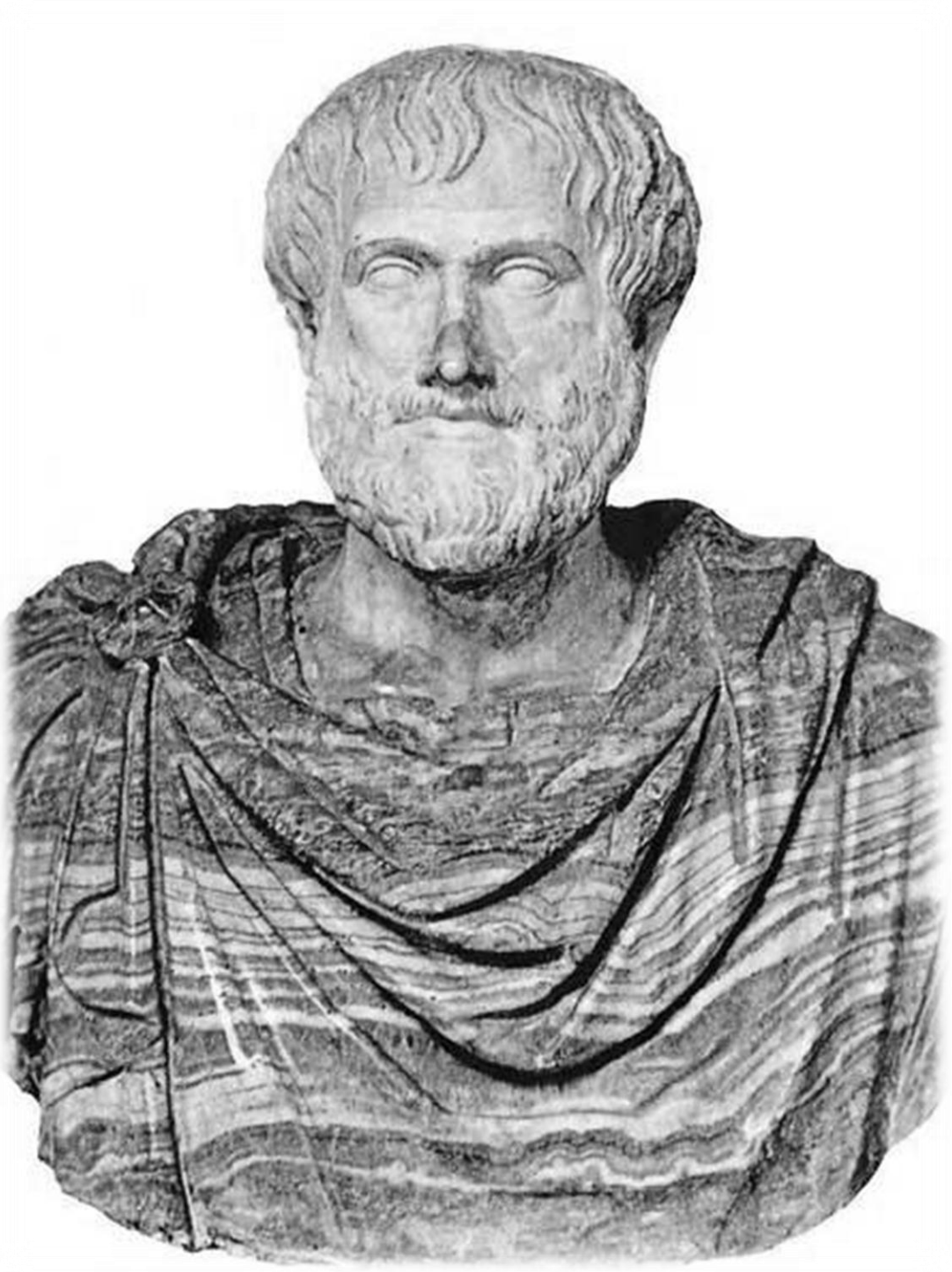
Trabajo de Diploma para optar por el Título de Ingeniero Informático

Autores: Dunier Bichot González

Yosbel Echevarría Millares.

Tutor: Raúl Velázquez Álvarez.

LA HABANA, JUNIO DE 2012



“El sabio no dice todo lo que piensa, pero siempre piensa todo lo que dice.”

Aristóteles
Filósofo griego
(384 AC - 322 AC)

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los **18** días del mes de **Junio** del año **2012**.

_____ _____

Dunier Bichot González Yosbel Echevarría Millares

_____

Ing. Raúl Velázquez Álvarez

Agradecimientos:

Agradezco primeramente a mi mamá y a mi abuela por su apoyo incondicional.

A mi novia por toda su dedicación.

A mi papá, a mi hermano, a mi madrastra y a mi padrastro.

A mis familiares, amistades y compañeros de aula.

A mi compañero de tesis.

A mi tutor.

Dunier Bichot González

Agradecimientos:

Agradezco a Jesucristo por todo lo que soy.

A mi padre que siempre quiso que fuera universitario y creyó en mí.

A mi esposa y su familia por su apoyo incondicional.

A todos mis familiares, amigos y compañeros de clase.

A mi compañero de tesis.

A mi tutor.

Yosbel Echevarría Millares

Dedicatoria:

A mi familia, en especial a mi mamá Letty F. Florian González

y a mi abuela Anastacia Florian Terrero.

Dunier Bichot González

A la memoria de mi abuela

Eufemia Alonso Pérez

“Mamá”

(1918-2005)

Yosbel Echevarría Millares

Resumen:

Los sistemas de gobierno electrónico requieren en su desempeño de un alto grado de seguridad; dado que ningún sistema puede considerarse 100% seguro, las acciones han de encaminarse a evitar al máximo la ocurrencia de fallas, faltas y errores.

Los estándares y modelos de calidad de software guían la forma en que se aplica la ingeniería de software. En el desarrollo de un sistema informático es necesario que la organización, para asegurar la calidad, no solo vele por la calidad del producto, sino que tenga además un enfoque hacia la calidad de sus procesos.

En el presente trabajo se hace la propuesta de una guía para el aseguramiento de la seguridad en los sistemas de gobierno electrónico, contribuyendo a reducir vulnerabilidades en los mismos y con ello elevar la calidad del software.

A través de encuestas a arquitectos y líderes de proyectos del Centro de Gobierno Electrónico, CEGEL, así como a especialistas de seguridad del Centro Nacional de Calidad de Software CALISOFT, se logró un acercamiento a la problemática.

La guía propuesta define un conjunto de actividades en las distintas etapas del ciclo de vida de desarrollo de software, teniendo en cuenta elementos críticos definidos por los autores en cuanto a la seguridad como una característica de calidad de software. Esta estrategia se ha validado positivamente con especialistas en el tema, que han dado un criterio acertado sobre la misma, lo que da la posibilidad de que pueda ser aplicable.

Palabras Claves:

Aseguramiento, Calidad, Estándar, Guía, Modelo, Seguridad, Software.

Abstract

The e-government systems require in their fulfillment a high level of security; as any system can be considered 100% sure, the actions must be guided to avoid any failure or mistakes, the models an standards guide the way in which software engineering is applied and they have the aim to produce the high quality software. Organization to ensure the quality is needed in the development of an informatics system, not only because of the product quality but also it should be directed towards the quality of its processes.

In this research a proposal of guide is made to ensure the security in the e-government systems, helping to reduce vulnerabilities in them and thus to elevate the quality of the software.

An approach to this topic was achieved through inquires to architects and leaders of projects from CEGEL, and also to security specialist from CALISOFT.

The proposed guide defines a group of activities in the different steps of the software development life cycle, taking into account critical elements defined by the authors according to security as a quality characteristic of software. This strategy has been validated positively with specialists who have provided a proper criterion about it which brings on the possibility that it can be applied.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Cada día los sistemas informáticos se tornan más extensos y complejos, por lo que se hace necesario el estudio de la seguridad del software como un elemento crucial durante todas las etapas de desarrollo del mismo. En la actualidad se desarrollan sistemas informáticos cada vez más críticos, que automatizan aspectos muy importantes del negocio, en los que cualquier defecto provoca un mayor impacto negativo en el cliente.

El activo más importante con que se cuenta es la información, por tanto deben existir técnicas, más allá de la seguridad física que la asegure. Debido a esto, la seguridad lógica toma un papel fundamental, ya que la mayoría de los daños que puede sufrir la organización que posee el sistema, no será sobre los medios físicos, sino contra la información por él almacenada y procesada. Debe evitarse por todos los medios cualquier posible falla de la seguridad de los mismos.

Los sistemas de gobierno electrónico requieren en su desempeño de un alto grado de seguridad, dado que ningún sistema puede considerarse 100% seguro, las acciones han de encaminarse a evitar al máximo la ocurrencia de fallas, faltas y errores. La información que en estos se maneja es de gran importancia, su indebida manipulación puede acarrear grandes perjuicios. Se hace necesaria la implementación de un entorno seguro de desarrollo que contribuya a la seguridad lógica.

En el marco de la informatización del país y en la búsqueda de respuesta a las necesidades de la gestión pública, basada en la utilización de las Tecnologías de la Información y de las Comunicaciones (TIC), existen en Cuba proyectos con el fin de lograr que el desarrollo de software y servicios informáticos, se conviertan en aliados necesarios. Uno de estos proyectos es la Universidad de las Ciencias Informáticas (UCI), creada en el año 2002 para potenciar y fomentar el desarrollo de software dentro del país.

El Centro de Gobierno Electrónico (CEGEL), perteneciente a la UCI, es un centro de referencia nacional en el desarrollo de proyectos, servicios y soluciones informáticas para la gestión de las áreas de gobierno. El mismo tiene la misión de *"Satisfacer necesidades de clientes gubernamentales, mediante el desarrollo de productos, servicios y soluciones integrales de alta confiabilidad, calidad, competitividad, fidelidad y eficiencia, a partir de un personal altamente*

calificado". Actualmente CEGEL, como parte de sus líneas de investigación, desarrolla proyectos de sistemas de software para la informática jurídica. (Lang, 2011)

A partir de entrevistas realizadas a los proyectos CEGEL (Ver anexo1), se detectó que durante el proceso de aseguramiento de la calidad, no cuentan en la actualidad con un equipo de seguridad en los proyectos productivos que contribuya a reducir la incidencia de vulnerabilidades, desde el inicio y durante las distintas etapas de desarrollo del software. No se realizan pruebas de seguridad que detecten a tiempo las posibles fallas de seguridad. La detección de vulnerabilidades en etapas tardías, provoca una demora considerable en su fecha de entrega, al tener que implementar estrategias para mitigar estas brechas de seguridad. No tienen definidos estándares y normas de seguridad en el plan de aseguramiento de la calidad del proyecto. Se hace muy difícil realizar pruebas a aplicaciones de escritorio, debido al desconocimiento sobre estas, así como las dificultades que estas provocan al interactuar estrechamente con el sistema operativo.

Las principales vulnerabilidades que se han detectado en aplicaciones web, durante las pruebas de calidad realizadas a los proyectos en la universidad han sido:

- Revelación de información a través del banner.
- Autocompletamiento habilitado.
- Gestión de sesiones.
- Brechas del phpMyAdmin.
- Mostrar la contraseña en texto plano.
- Cross-site Scripting.
- Permisos para adjuntar archivos ejecutables.
- Pykto plugin.
- Path Disclosure.
- Existencia de muchos puertos abiertos.

PROBLEMA A RESOLVER

¿Cómo contribuir al aseguramiento de la seguridad lógica en los sistemas informáticos de gobierno electrónico, de manera que favorezca la reducción de las vulnerabilidades en los mismos?

OBJETO DE ESTUDIO

Aseguramiento de la seguridad lógica.

OBJETIVO GENERAL

Desarrollar una guía para el aseguramiento de la seguridad lógica, que permita reducir las vulnerabilidades en el desarrollo de los sistemas informáticos de gobierno electrónico.

CAMPO DE ACCIÓN

La seguridad lógica en los sistemas informáticos de gobierno electrónico.

IDEA A DEFENDER

Con el desarrollo de una guía para el aseguramiento de la seguridad lógica, en el desarrollo de sistemas informáticos de gobierno electrónico, se contribuye a reducir las vulnerabilidades de los mismos.

OBJETIVOS ESPECÍFICOS

1. Elaborar el marco teórico de la investigación.
2. Desarrollar la propuesta de solución.
3. Validar la solución propuesta.

TAREAS DE LA INVESTIGACIÓN

- Realización de un estudio del estado del arte de la temática en Cuba y el mundo.
- Análisis del estándar (ISO/IEC 25010), así como otros modelos y estándares internacionales, que traten el tema de las características de calidad de software.
- Análisis de las subcaracterísticas existentes en dependencia del estudio de los estándares y modelos definidos internacionalmente, que traten el tema de las características de calidad de software.
- Análisis de los diferentes atributos existentes que conforman cada subcaracterística.

- Definición de los atributos necesarios durante el ciclo de desarrollo, basado en el programa de mejora de la UCI que plantea CMMI, para que las aplicaciones de escritorio de gobierno electrónico cumplan con la característica de seguridad.
- Definición de acciones o actividades a realizar a lo largo del ciclo de desarrollo de software, para asegurar la seguridad en el producto final, basándose en los elementos anteriormente definidos.
- Realización de pruebas a los sistemas del centro a partir de los elementos definidos en la solución propuesta para constatar la existencia de vulnerabilidades.

MÉTODOS DE INVESTIGACIÓN

MÉTODOS TEÓRICOS

ANALÍTICO-SINTÉTICO

Permite realizar un estudio detallado de las teorías, tendencias y documentos relacionados con la calidad del software y la seguridad de sistemas informáticos, para comprender la importancia que tiene la seguridad actualmente y su tratamiento. Se sintetizan los elementos más importantes y de mayor utilidad para el desarrollo de la investigación.

HISTÓRICO LÓGICO

Permite constatar teóricamente cómo ha evolucionado la seguridad en los sistemas informáticos, así como la necesidad imperante que existe de aplicar buenas estrategias para obtener productos de software seguros debido a la complejidad de estos.

MÉTODOS EMPÍRICOS

ENTREVISTA

Se realizaron entrevistas a arquitectos y líderes de proyectos productivos de CEGEL, así como a especialistas de seguridad informática de CALISOFT, con el objetivo de obtener información acerca de cómo se trabaja y se sigue la seguridad en los sistemas informáticos en la universidad. Se pudo percibir los tipos de pruebas que se realizan y las herramientas que se emplean para las mismas.

ENCUESTA

Facilita conocer la opinión y valoración de especialistas seleccionados, para corroborar que los elementos definidos en la propuesta pueden influir en la seguridad de los sistemas que se desarrollen en el centro.

La presente investigación posee una estructura de tres capítulos, enmarcando la información por especialidades y temas, facilitando su consulta y entendimiento. A continuación se ofrece una breve descripción de cada uno de ellos:

Capítulo I Fundamentación teórica: En este capítulo se presentan las posiciones teóricas en torno a la calidad, aseguramiento de la calidad, seguridad, seguridad lógica, principales vulnerabilidades y técnicas para el aseguramiento de la seguridad de un software. Se realiza un análisis de los modelos y estándares de calidad.

Capítulo II Propuesta de solución: En este capítulo se describe una guía para el aseguramiento de la seguridad lógica en los sistemas informáticos de gobierno electrónico según el ciclo de vida definido en el programa de mejora en el que está encaminada la UCI, basado en el nivel 2 de CMMI.

Capítulo III Validación de la propuesta: En este capítulo se valida la propuesta. Se muestra un análisis estadístico de los resultados de encuestas realizadas a especialistas, con el objetivo de realizar una validación de la propuesta.

CAPÍTULO 1

1.1 INTRODUCCIÓN

En este capítulo se presentan los conceptos de calidad de software, aseguramiento de la calidad, seguridad y seguridad lógica. Se realiza una caracterización de los modelos y estándares de calidad de software existentes, para definir aspectos importantes y necesarios que permitan un mayor entendimiento de la seguridad como una característica de calidad de software.

1.2 CALIDAD DE SOFTWARE

La calidad de los productos de software es uno de los temas más trascendentales en la actualidad debido a la repercusión de la misma en el desarrollo de productos informáticos. El interés por la calidad crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar productos poco fiables o que realmente no dan respuesta a sus necesidades.

¿Qué es la calidad del software?

> *"Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor."* (RAE, 2010)
>
> Diccionario Cervantes de la Real Academia de la Lengua Española.

> *"Cero defectos" ó "Conformidad con los Requisitos"* (Oyarzún L., 2006)
>
> Philip Bayard Crosby, empresario norteamericano, autor que contribuyó a la teoría gerencial y a las prácticas de la gestión de la calidad.

> *"*La calidad es satisfacción del cliente*."* (Espino, 2011)
>
> William Edwards Deming, profesor universitario, autor de textos, consultor y difusor del concepto de calidad total.

"Adecuación (del producto o servicio) al uso." (Oyarzún L., 2006)

Joseph Moses Juran, consultor de gestión, es principalmente recordado como un evangelista de la calidad, la gestión de la calidad y la escritura de varios libros influyentes sobre esos temas.

"La calidad como resultado de la interacción de dos dimensiones: Dimensión subjetiva (lo que el cliente quiere) y dimensión objetiva (lo que se ofrece)." (Espino, 2011)

Walter A Shewhart, físico, ingeniero y estadístico estadounidense conocido como el padre del control estadístico de la calidad.

"El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas." (Eickelman, 2004)

Instituto de Ingenieros Eléctricos y Electrónicos por sus siglas en ingles IEEE, asociación técnico-profesional internacional sin ánimos de lucro.

"Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente. (Pressman, 2002)

Dr. Roger S. Pressman, autoridad internacionalmente reconocida en el campo de la ingeniería de software con más de tres décadas de experiencia.

Tomando la definición de Pressman, se puede decir que los requisitos del software sirven de base para medir la calidad de un software. Es vital en la construcción del software la adopción de metodologías o estándares que contribuyan al aseguramiento de la calidad.

"Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Si no se sigue ninguna metodología siempre habrá falta de calidad. Todas las metodologías y herramientas tienen un único fin producir software de alta calidad." (Scalone, 2006)

Es por ello que debe tenerse en cuenta todo lo que pueda afectar el desarrollo del software desde etapas tempranas y a lo largo de todo el ciclo de vida del mismo, de esta manera asegurar la calidad del producto final.

1.3 ASEGURAMIENTO DE LA CALIDAD

El aseguramiento de la calidad de software se realiza de forma independiente al equipo de desarrollo, integra la planificación, estimación y supervisión del proceso de desarrollo del software.

> *"El Aseguramiento de Calidad del Software es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el software satisfará los requisitos dados de calidad."* (Pressman, 2002)

Antes de iniciar el desarrollo de una aplicación es necesario diseñar el aseguramiento de la calidad, no después.

> *"El aseguramiento de la calidad del software engloba: Un enfoque de gestión de calidad, métodos y herramientas de Ingeniería del Software, revisiones técnicas formales aplicables en el proceso de software, una estrategia de prueba multiescala, el control de la documentación del software y de los cambios realizados, procedimientos para ajustarse a los estándares de desarrollo del software y mecanismos de medición y de generación de informes."* (Scalone, 2006)

El aseguramiento de calidad del software tiene como misión esencial garantizar los requisitos de calidad, no solo desde el punto de vista funcional, sino también la eficiencia, mantenimiento, fiabilidad, portabilidad, seguridad, arquitectura, documentación, infraestructura tecnológica, código generado etc.

1.4 SEGURIDAD

> *"La Seguridad, en pocas palabras, es asegurar la confidencialidad, integridad, y disponibilidad de sus sistemas y redes. La seguridad se divide en seguridad física y seguridad lógica."* (Huerta, 2000)

También se conoce como seguridad un estado de cualquier sistema (informático o no), que indica que está libre de peligro, daño o riesgo. Se entiende como peligro o daño, todo aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo. Para la mayoría de los expertos el concepto de seguridad en la informática es utópico porque no existe un sistema 100% seguro. Para que un sistema se pueda definir como seguro debe cumplir con los tres pilares de la seguridad:

- **Integridad**: La información sólo puede ser modificada por quien está autorizado.
- **Confidencialidad**: La información sólo debe ser legible para los autorizados.
- **Disponibilidad**: La información sólo debe estar disponible cuando se necesita.

Así como otras dos características importantes para algunos proyectos en específico que las necesiten como:

- **No repudio:** No se pueda negar la autoría.
- **Autenticidad:** Informa que el archivo es el auténtico.

1.5 SEGURIDAD LÓGICA

Se han dado varios conceptos acerca de qué es la seguridad lógica, pero al final todas convergen en la misma definición:

> *La seguridad lógica consiste en la "aplicación de barreras y procedimientos que resguarden el acceso a los datos y sólo se permita acceder a ellos a las personas autorizadas para hacerlo."* (Chan Basto, 2010)

El Instituto Nacional de Normas y Tecnología, NIST por sus siglas en inglés, es una agencia de la Administración de Tecnología del Departamento de Comercio de los Estados Unidos, la cual ha resumido los siguientes requisitos mínimos de seguridad para cualquier sistema.

- **Identificación y autentificación:** Se denomina identificación al momento en que el usuario se da a conocer en el sistema y autenticación a la verificación que realiza el sistema sobre esta identificación.

- **Roles:** El acceso a la información también puede controlarse a través de la función o rol del usuario que requiere dicho acceso.

- **Transacciones:** Se implementan controles a través de las transacciones, por ejemplo solicitando una clave al requerir el procesamiento de una transacción determinada.

- **Limitaciones a los servicios:** Se refieren a las restricciones que dependen de parámetros propios de la utilización de la aplicación o preestablecidos por el administrador del sistema. Un ejemplo podría ser que en la organización se disponga de licencias para la utilización simultánea de un determinado producto de software para cinco personas, en donde exista un control de sistema que no permita la utilización del producto a un sexto usuario.

- **Modalidad de acceso:** Se refiere al modo de acceso que se permite al usuario sobre los recursos y a la información. Esta modalidad puede ser: lectura, escritura, ejecución, borrado, total.

- **Ubicación y horario:** El acceso a determinados recursos del sistema puede estar basado en la ubicación física o lógica de los datos o personas. En cuanto a los horarios, este tipo de controles permite limitar el acceso de los usuarios a determinadas horas de día o a determinados días de la semana.

- **Control de acceso interno:** Se refiere principalmente a:
    - Palabras claves (passwords): Generalmente se utilizan para realizar la autenticación del usuario y sirven para proteger los datos y aplicaciones.
    - Encriptación: La información encriptada solamente puede ser desencriptada por quienes posean la clave apropiada.
    - Listas de control de accesos: Se refiere a un registro donde se encuentran los nombres de los usuarios que obtuvieron el permiso de acceso a un determinado recurso del sistema, así como la modalidad de acceso permitido.

Control de acceso externo: Se refiere principalmente a:

- Dispositivos de control de puertos: Estos dispositivos autorizan el acceso a un puerto determinado.
- Firewalls o puertas de seguridad: Permiten bloquear o filtrar el acceso entre dos redes, usualmente una privada y otra externa.
- Acceso de personal contratado: Debido a que este tipo de personal en general presta servicios temporarios, debe ponerse especial consideración en la política y administración de sus perfiles de acceso.

- **Administración:** Se refiere a la organización del personal y usuarios, este proceso lleva generalmente cuatro pasos:
  - Definición de puestos: Debe contemplarse la máxima separación de funciones posibles y el otorgamiento del mínimo permiso de acceso requerido por cada puesto para la ejecución de las tareas asignadas.
  - Determinación de la sensibilidad del puesto: Para esto es necesario determinar si la función requiere permisos riesgosos que le permitan alterar procesos, perpetrar fraudes o visualizar información confidencial.
  - Elección de la persona para cada puesto: Requiere considerar los requisitos de experiencia y conocimientos técnicos necesarios para cada puesto. Para los puestos definidos como críticos puede requerirse una verificación de los antecedentes personales.
  - Entrenamiento inicial y continuo del empleado: Cuando la persona seleccionada ingresa a la organización, además de sus responsabilidades individuales para la ejecución de las tares que se asignen, deben comunicárseles las políticas organizacionales, haciendo hincapié en la política de seguridad. (NIST, 2010)

1.5.1 TÉCNICAS PARA ASEGURAR EL SISTEMA

- **Codificar la información:** Criptología, criptografía y criptociencia, contraseñas difíciles de averiguar a partir de datos personales del individuo.

- **Validación de la entrada y salida de información:** La entrada y salida de información es el principal mecanismo que dispone un atacante para enviar o recibir código

malicioso contra el sistema. Por tanto, siempre debe verificarse que cualquier dato entrante o saliente es apropiado y en el formato que se espera. Las características de estos datos deben estar predefinidas y deben verificarse en todas las ocasiones.

- **Vigilancia de red:** Establecer una zona desmilitarizada (DMZ), cuando algunas máquinas de la red interna deben ser accesibles desde una red externa (servidores web, servidores de correo electrónico, servidores FTP), se crea una nueva interfaz hacia una red separada a la que se pueda acceder tanto desde la red interna como por vía externa sin correr el riesgo de comprometer la seguridad de la compañía.

- **Tecnologías repelentes o protectoras:** Cortafuegos, sistema de detección de intrusos, antispyware, antivirus, llaves para protección de software, etc. Mantener los sistemas de información con las actualizaciones que más impacten en la seguridad.

- **Sistema de respaldo remoto:** Servicio remoto de salvas.

- **Utilización y reutilización de componentes de confianza:** Debe evitarse reinventar y desarrollar código constantemente. Por tanto, cuando exista un componente que resuelva un problema de forma correcta, lo más inteligente es documentarlo y reutilizarlo.

- **Defensa en profundidad:** Nunca confiar en que un componente realizará su función de forma permanente y ante cualquier situación. Se ha de disponer de los mecanismos de seguridad suficientes para que cuando un componente del sistema falle ante un determinado evento, otros sean capaces de detectarlo.

- **Ofrecer la mínima información:** Ante una situación de error o una validación negativa, los mecanismos de seguridad deben diseñarse para que faciliten la mínima información posible. De la misma forma, estos mecanismos deben estar diseñados para que una vez denegada una operación, cualquier operación posterior sea igualmente denegada. (blogspot, 2010)

1.5.2 PRINCIPALES VULNERABILIDADES

Las principales vulnerabilidades que se han detectado en aplicaciones web, durante las pruebas de calidad realizadas a los proyectos en la universidad se describen a continuación:

- **Revelación de información a través del banner:** La aplicación revela su tipo y versión a través del banner cuando da algún error.

- **Autocompletamiento habilitado:** Se deja habilitado el autocompletamiento en los campos de texto, revelando información como: usuario, datos personales, entre otros; a personas no autorizadas.

- **Gestión de sesiones:** Después que el usuario cierra la sesión en el sistema, el próximo usuario selecciona ir atrás a la aplicación y vuelve a abrirse la sesión anterior, sin autenticación.

- **Brechas del phpMyAdmin**: Instalan phpMyAdmin y todas las vulnerabilidades que este presenta, pasan a afectar la seguridad del software que se desarrolla.

- **Mostrar contraseña en texto plano**: La contraseña viaja desde que se escribe hasta la base de datos en formato plano, sin cifrar, pudiendo ser capturada en ese transcurso.

- **Cross-site Scripting:** Falta de mecanismos de filtrado en los datos de entrada que permiten el ingreso y envió de datos sin validación, aceptando scripts completos, pudiendo generar secuencias de comandos maliciosas. Existen dos tipos uno vía correo y otro en la web.

- **Permisos para adjuntar archivos ejecutables:** En campos donde el usuario puede adjuntar documentos y subir información, se encuentran habilitados los permisos para subir archivos ejecutables que pueden ser malignos.

- **Pykto plugin:** Al modificar campos de la URL, accedes a datos que no deberías ver o tener acceso a ellos.

- **Path Disclosure:** Tras insertar datos erróneos en la dirección URL, se muestra un error que deja al descubierto la ruta de la aplicación del script en ejecución.

- **Existencia de muchos puertos abiertos:** Tanto los usuarios legítimos como los atacantes acceden al sistema a través de los puertos abiertos. Cuantos más puertos existan abiertos, más posibilidades de que alguien pueda conectar con el sistema.

1.6 MODELOS Y ESTÁNDARES INTERNACIONALES PARA LAS CARACTERÍSTICAS DE CALIDAD DE SOFTWARE

Hoy en día existen empresas que cuentan con modelos y estándares de calidad que benefician su trabajo en el desarrollo de software seguro.

La Organización Internacional para la Estandarización y la Comisión Electrotécnica Internacional ISO/IEC por sus siglas en inglés, definen un estándar como:

> *"Se trata de un conjunto de reglas que se usan de manera habitual como los buenos principios, prácticas o directrices. Un estándar controla cómo las personas deben desarrollar y gestionar los materiales, productos, servicios, tecnologías, procesos y sistemas."* (Demand Media, 2012)

Según Ernesto Quiñones A. desarrollador y director de proyectos de desarrollo de software, especializado en sistemas de información un modelo es:

> *"Es un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos."* (Quiñones A., 2006)

1.6.1 MODELO MCCALL (1977)

El modelo McCall organiza los factores en tres ejes o puntos de vista, desde los cuales el usuario puede contemplar la calidad, operación, revisión, y transición del producto. Cada punto de vista se descompone en una serie de factores que determinan la calidad de cada uno de ellos. Cada factor determinante de la calidad, se descompone a su vez, en una serie de criterios o propiedades que determinan su calidad. Los criterios pueden ser evaluados mediante un conjunto de métricas, deben fijarse para cada uno, valores máximo y mínimo aceptables.

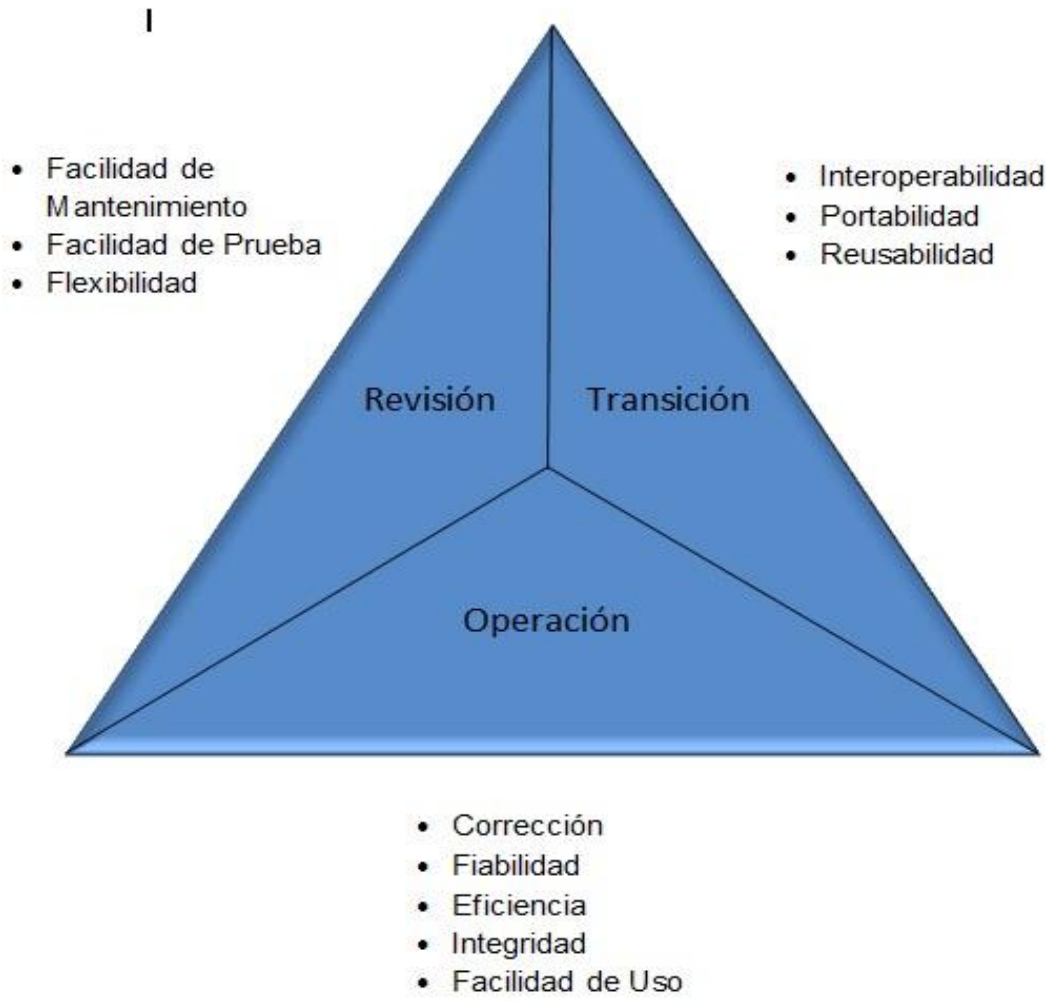


Figura 1. Factores de Calidad del Modelo McCall

El modelo McCall se basa en 11 factores de calidad, que se organizan de la siguiente manera:

Factor	Criterio
Correctitud	✓ Rastreabilidad ✓ Completitud ✓ Consistencia
Confiabilidad	✓ Consistencia ✓ Exactitud ✓ Tolerancia a fallas
Eficiencia	✓ Eficiencia de ejecución ✓ Eficiencia de almacenamiento
Integridad	✓ Control de acceso ✓ Auditoría de acceso
Usabilidad	✓ Operabilidad ✓ Entrenamiento ✓ Comunicación
Mantenibilidad	✓ Simplicidad ✓ Concreción
Capacidad de Prueba	✓ Simplicidad ✓ Instrumentación ✓ Auto-descriptividad ✓ Modularidad
Flexibilidad	✓ Auto-descriptividad ✓ Capacidad de expansión ✓ Generalidad ✓ Modularidad
Portabilidad	✓ Auto-descriptividad ✓ Independencia del sistema ✓ Independencia de máquina
Reusabilidad	✓ Auto-descriptividad ✓ Generalidad ✓ Modularidad ✓ Independencia del sistema ✓ Independencia de máquina
Interoperabilidad	✓ Modularidad ✓ Similitud de comunicación ✓ Similitud de datos.

Figura 2. Modelo de McCall. (Camacho, 2004)

El modelo de McCall trata la seguridad como un factor asociado a la integridad, donde especifica si el software cumple con este requisito de calidad. Este modelo no trata la seguridad como una característica independiente. Dentro de la integridad se definen tres aspectos significantes que son:

- **Control de accesos:** Atributos del software que proporcionan control de acceso al software y los datos que maneja.

- **Facilidad de auditoría:** Atributos del software que facilitan la auditoría de los accesos al software.

- **Seguridad:** La disponibilidad de mecanismos que controlen o protejan los programas o los datos.

1.6.2 MODELO FURPS (1987)

"El modelo FURPS propuesto por Robert Grady y Heweltt Packard Co (HP) cuenta con cinco características de calidad del software: funcionalidad, facilidad de uso, confiabilidad, rendimiento y facilidad de soporte." (Scalone, 2006)

Factor de Calidad	Atributos
Funcionalidad	✓ Características y capacidades del programa ✓ Generalidad de las funciones ✓ Seguridad del sistema
Facilidad de uso	✓ Factores humanos ✓ Factores estéticos ✓ Consistencia de la interfaz ✓ Documentación
Confiabilidad	✓ Frecuencia y severidad de las fallas ✓ Exactitud de las salidas ✓ Tiempo medio de fallos ✓ Capacidad de recuperación ante fallas ✓ Capacidad de predicción
Rendimiento	✓ Velocidad del procesamiento ✓ Tiempo de respuesta ✓ Consumo de recursos ✓ Rendimiento efectivo total ✓ Eficacia
Capacidad de Soporte	✓ Extensibilidad ✓ Adaptabilidad ✓ Capacidad de pruebas ✓ Capacidad de configuración ✓ Compatibilidad ✓ Requisitos de instalación

Figura 3. Modelo de FURPS. (Camacho, 2004)

El modelo FURPS tarta la seguridad como un factor asociado a la funcionalidad, donde especifica si el software cumple con este requisito de calidad. Este modelo no trata la seguridad como una característica independiente. Dentro de la funcionalidad se definen tres aspectos significantes que son:

- Características y capacidades del programa.
- Generalidades de las funciones.
- Seguridad del sistema.

1.6.3 ISO/IEC 9126-1: 2001 QUALITY MODEL

> *"El estándar ISO/IEC 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad para un producto de software."* (Pressman, 2002)

> *"Este estándar es una simplificación del Modelo McCall."* (Losavio, 2003)

Este estándar identifica seis características básicas de calidad, que pueden estar presentes en cualquier producto de software. El estándar de calidad interna y externa está formado por las siguientes características: funcionalidad, confiabilidad, facilidad de uso, eficiencia, facilidad de mantenimiento y portabilidad.

La norma ISO/IEC 9126-1 presenta la seguridad como una subcaracterística asociada a la funcionalidad, donde puntualiza si el software cumple con este requisito de calidad. Este estándar no trata la seguridad como una característica independiente.

Característica	Subcaracterística
Funcionalidad	✓ Adecuación ✓ Exactitud ✓ Interoperabilidad ✓ Seguridad
Confiabilidad	✓ Madurez ✓ Tolerancia a fallas ✓ Recuperabilidad
Usabilidad	✓ Entendibilidad ✓ Capacidad de aprendizaje ✓ Operabilidad
Eficiencia	✓ Comportamiento en tiempo ✓ Comportamiento de recursos
Mantenibilidad	✓ Analizabilidad ✓ Modificabilidad ✓ Estabilidad ✓ Capacidad de pruebas
Portabilidad	✓ Adaptabilidad ✓ Instalabilidad ✓ Reemplazabilidad

Figura 4. Modelo ISO/IEC 9126. (Camacho, 2004)

1.6.4 Estándar de calidad ISO/IEC 25010 SQuaRE

> *"El estándar SQuaRE (Software Quality Requirements and Evaluation) es una revisión de la ISO 9126-1. Hay dos aspectos importantes en el campo de la calidad del software, el producto y el proceso. SQuaRE se centra en el lado del producto. SQuaRE hereda el estándar de calidad de la ISO 9126-1."* (Morilla, 2009)

A continuación se muestran las ocho características, del estándar de calidad, que debe poseer todo desarrollo según la norma ISO/IEC 25010.

- Funcionalidad
- Seguridad
- Interoperabilidad
- Fiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad

El estudio de los estándares y modelos de calidad ha permitido profundizar en los beneficios que pueden obtenerse con la aplicación de los mismos, se determinó utilizar como referencia para esta investigación el estándar SQuaRE por lo novedoso y estructurado del mismo, por otra parte los demás modelos son antiguos y tratan la seguridad como una subcaracterística.

1.7 ESTÁNDAR DE CALIDAD ISO/IEC 25010 SQUARE

SQuaRE nace con el objetivo de responder a las necesidades de los usuarios a través de un conjunto de documentos unificados cubriendo tres procesos de calidad complementarios: especificación de requisitos, medidas y evaluación. Por lo tanto, SQuaRE se creó para satisfacer una serie de necesidades que existían con la ISO 9126 y la ISO/IEC 14598 pertenecientes a la primera generación de estándares de calidad de un producto software. Por consiguiente, SQuaRE pertenece a la segunda generación de calidad de un producto software. (García, 2009)

El estándar de calidad SQuaRE categoriza la calidad del software en características, las cuales están divididas en subcaracterísticas. Existen tres versiones de SQuaRE:

- ISO/IEC 25010 Mayo 2007.
- ISO/IEC 25010 Julio 2007.
- ISO/IEC 25010 Julio 2008.

SQuaRE maneja la calidad del producto en las tres distintas etapas dentro del ciclo de vida de un producto:

- La calidad interna trata de productos en desarrollo.
- La calidad externa de productos en funcionamiento.
- La calidad en uso se refiere a productos en uso.

Requisitos de calidad interna: Especifican las propiedades de productos software intermedios, como por ejemplo módulos o clases. Se utilizan para verificar el producto a lo largo de las

distintas etapas del desarrollo y pueden utilizarse también para definir estrategias y criterios de evaluación y verificación.

Requisitos de calidad externa: Especifican los umbrales de aceptación de medidas externas y se utilizan para la verificación y validación técnica del producto. Estos requisitos ayudan a determinar los requisitos de calidad interna pero además, pueden servir para predecir si se alcanzará la calidad en uso deseada.

Requisitos de calidad en uso: Especifican los requisitos desde el punto de vista del usuario. Estos requisitos son los que determinan la validación del software por parte del usuario. Como indica el ciclo de vida, la especificación de requisitos de calidad en uso ayuda a determinar los requisitos de calidad externa.

Calidad del software (interna y externa)							
Adecuación funcional	***Fiabilidad***	***Eficiencia de rendimiento***	***Operabilidad***	***Seguridad***	***Compatibilidad***	***Mantenibilidad***	***Transmisibilidad***
Adecuación	*Disponibilidad*	*Tiempo de respuesta*	*Reconoc. de adecuación*	*Confidencialidad*	*Capacidad a reemplazo*	*Modularidad*	*Portabilidad*
Precisión	*Tolerancia a fallos*	*Utilización de recursos*	*Capacidad de aprendizaje*	*Integridad*	*Capacidad a coexistencia*	*Reusabilidad*	*Adaptabilidad*
Adherencia a normas	*Recuperación*	*Adherencia a normas*	*Facilidad de uso*	*No rechazo*	*Interoperabilidad*	*Capacidad a análisis*	*Capacidad a instalación*
	Adherencia a normas		*Util*	*Responsabilidad*	*Adherencia a normas*	*Capacidad a cambios*	*Adherencia a normas*
			Atractivo	*Autenticidad*		*Estable modificación*	
			Accesible técnicamente	*Adherencia a normas*		*Capacidad a testing*	
			Adherencia a normas			*Adherencia a normas*	

Figura 5. Modelo de calidad para la calidad interna y externa para la versión ISO/IEC 25010 Julio 2008. (Morilla, 2009)

Calidad de uso		
Usabilidad de uso	**Flexibilidad de uso**	**Seguridad de uso(riesgo humano)**
Efectividad de uso	Conformidad de contexto de uso	Seguridad y salud del operador
Eficiencia de uso	Extensión de contexto de uso	Salud y seguridad Pública
Satisfacción de uso	Accesibilidad de uso	Daño al entorno de uso
Adherencia a normas	Adherencia a normas	Daños comerciales de uso
		Adherencia a normas

Figura 6. Modelo de Calidad para la calidad en Uso para la versión ISO/IEC 25010 Julio 2008.

(Morilla, 2009)

Los mayores beneficios de la serie SQuaRE sobre sus predecesores estándares incluyen:

- La coordinación de dirección sobre la medida y evaluación de calidad del producto software.
- Dirección para la especificación de requisitos de calidad del producto software.
- Armonización con ISO/IEC 15939 en forma de modelo de referencia de modelo de calidad presentado en el estándar SQuaRE.

La norma ISO/IEC 25010 está enfocada hacia la calidad del producto, pero para asegurar la calidad, es necesario que la organización tenga además un enfoque hacia la calidad de sus procesos.

1.8 Modelo Integrado de Capacidad y Madurez (CMMI)

En la actualidad la UCI está acometiendo un programa de mejora de sus procesos basado en el Modelo Integrado de Capacidad y Madurez (CMMI, por sus siglas en inglés) y con la contratación de los servicios de consultoría del SIE Center (Software Industry Excellence Center) del Tecnológico de Monterrey. El proceso de mejora está encaminado a que la universidad tenga una certificación internacional del nivel 2 de este modelo, hecho que la convertiría en la primera empresa cubana certificada con el mismo.

CMMI es un modelo de madurez de mejora de los procesos para el desarrollo de productos y servicios. Consiste en las mejores prácticas que tratan las actividades de desarrollo y de mantenimiento que cubren el ciclo de vida del producto, desde la concepción hasta la entrega y mantenimiento. (Beth Chrissis, 2009)

Entre las características de CMMI se encuentran:

- Eliminar inconsistencias.
- Reducir duplicaciones.
- Incrementar la claridad y comprensión.
- Proporcionar terminología común.
- Proporcionar estilos consistentes.
- Establecer reglas de construcción uniformes.
- Mantener componentes comunes. (Bertone, 2010)

Este modelo para software establece 5 niveles de madurez para clasificar a las organizaciones. Para que una organización se encuentre en un determinado nivel es necesario cumplir con todas las actividades definidas para ese nivel y para los niveles anteriores.

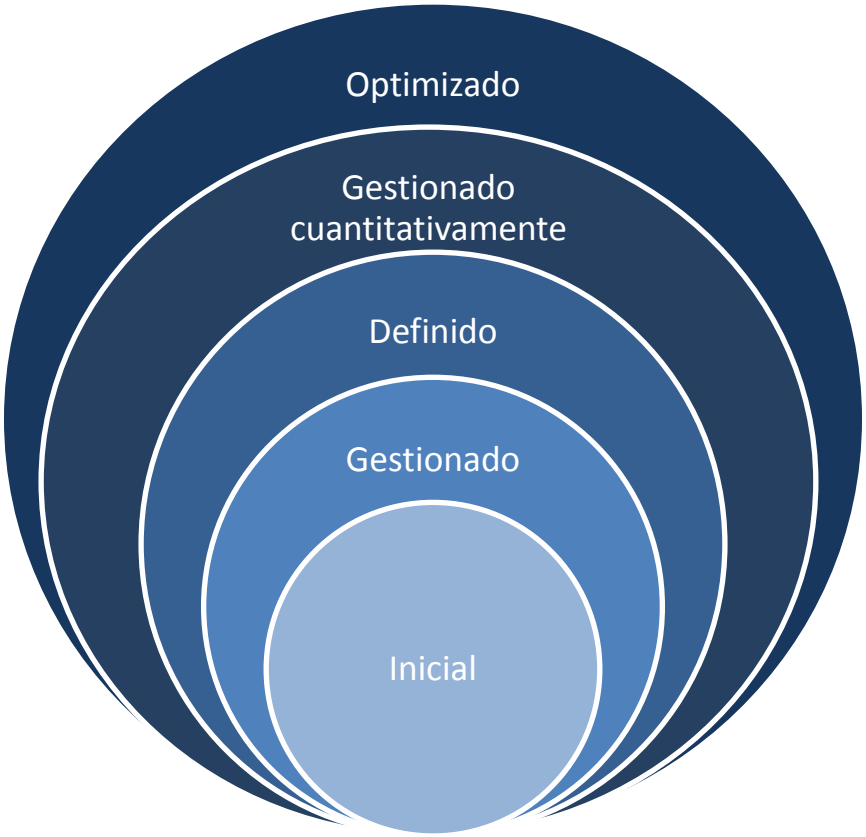


Figura 7. Niveles de CMMI.

Actualmente los proyectos certificados en la universidad se encuentran en el nivel 2 que cuenta con 7 áreas de procesos, que son: Administración de Acuerdo con Proveedores (SAM), Aseguramiento de la Calidad de Procesos y Productos (PPQA), Administración de la Configuración (CM), Planeación del Proyecto (PP), Monitoreo y Control de Proyecto (PMC), Administración de Requisitos (REQM) y Medición y Análisis (MA).

1.9 CONCLUSIONES PARCIALES

En el presente capítulo se realizó un análisis de los conceptos de calidad y aseguramiento de la calidad, escogiendo el más acorde a la investigación. Se describieron los aspectos fundamentales referentes a la seguridad lógica de los sistemas informáticos. En cuanto al enfoque de calidad de producto, se seleccionó la norma ISO/IEC 25010 SQuaRE como guía para el desarrollo de la propuesta de solución, luego de un estudio de los modelos de calidad existentes, por ser la más relevante con respecto a la seguridad del software. El programa de mejoras de la universidad está basado en el modelo CMMI nivel 2 y plantea un desarrollo estructurado por etapas, lo que permitirá determinar los elementos críticos en cuanto a la seguridad lógica durante todas las etapas de desarrollo del software.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1 INTRODUCCIÓN

En el siguiente capítulo se muestra la guía para el aseguramiento de la seguridad lógica en los sistemas informáticos de gobierno electrónico, basado en el ciclo de vida de los proyectos definidos en el nivel 2 de CMMI, como parte del proceso de mejora al que está encaminada la UCI. La carencia de una guía para el aseguramiento de la seguridad en el desarrollo de software fue lo que conllevó a esta propuesta de solución, que contribuirá a la calidad, con una mejora de la seguridad lógica en los proyectos del centro.

2.2 CARACTERÍSTICAS DE LOS PROYECTOS DEL CENTRO DE GOBIERNO ELECTRÓNICO

El CEGEL se organiza en dos departamentos, el de Informática Jurídica (IJ) y el de Gestión Gubernamental (GG). Estos desarrollan sus soluciones empleando para el tratamiento de la seguridad a los frameworks de java: Spring, Hibernate; de Microsoft .NET y de PHP: Symphony. Los proyectos de CEGEL desarrollan sus soluciones siguiendo el ciclo de vida definido por el programa de mejora que se lleva a cabo en la UCI.

En cada proyecto se deben tener en cuenta las características propias del software, según las necesidades del cliente, debido a que estos definen qué nivel de seguridad requerirá su sistema de acuerdo a la información que en él se procese. Los sistemas informáticos de gobierno electrónico son desarrollados para entidades con gran cantidad de información, en su mayoría con un alto grado de confidencialidad, lo que puede atentar contra la seguridad sino se tratan de manera adecuada.

2.3 CICLO DE VIDA DEL PROCESO DE DESARROLLO DE SOFTWARE BASADO EN EL PROGRAMA DE MEJORA. ELEMENTOS DEFINIDOS POR ETAPAS

La UCI está atravesando por un programa de mejora de sus procesos, por lo que se hace necesario realizar un estudio del ciclo de vida del proceso de desarrollo de software. Es necesario dar seguimiento a las actividades que hay realizar por etapas, para lograr un aseguramiento de la seguridad.

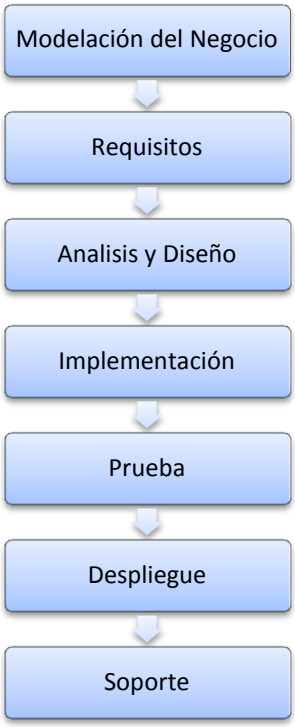


Figura 8. Etapas del ciclo de desarrollo del software según el programa de mejoras.

A continuación se muestra una descripción de las etapas y una serie de elementos críticos definidos por los autores para esta guía.

2.3.1 ESTUDIO PRELIMINAR

En esta etapa efectúan las actividades relacionadas con la planeación del proyecto. Además se realiza un estudio inicial del cliente que permite obtener información fundamental acerca del alcance del proyecto y realizar estimaciones de tiempo, esfuerzo y costo.

2.3.2 MODELACIÓN DEL NEGOCIO

El modelado del negocio es la etapa destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea automatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para la descripción y modelado de negocio pueden ser utilizadas diferentes técnicas como el modelado de casos de uso del negocio y la Notación para el Modelado de Procesos de Negocio, BPMN por sus siglas en ingles.

ELEMENTOS CRÍTICOS DE LA ETAPA

- **Modelo de casos de uso del negocio**: Se identifican correctamente los roles y responsabilidades de los actores del negocio asociados a los procesos del mismo.

- **Reglas del negocio:** Especifica cómo el negocio, incluyendo sus herramientas, debe funcionar, es descrito de manera formal como restricciones y derivaciones.

- **Documento de la Arquitectura:** El enfoque de la arquitectura del sistema haciendo uso de los estilos y patrones, para así poder ilustrar las características más importantes del sistema.

    - **Herramientas horizontales**
        - **Repositorios:** Donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.
        - **Control de versiones:** La gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.

2.3.3 REQUISITOS

El esfuerzo principal en la etapa de requisito es desarrollar un modelo del sistema que se va a construir. Incluye un conjunto de casos de uso, servicios que describen todas las interacciones que tendrán los usuarios con el software, estos responden a los requisitos funcionales del sistema. Además la especificación de requisitos incluye requisitos no funcionales.

ELEMENTOS CRÍTICOS DE LA ETAPA

- **Requisitos no funcionales:** Son propiedades o cualidades que el producto debe tener.

    - **Requisitos de seguridad**: La seguridad es tratada en varios aspectos (integridad, confidencialidad, disponibilidad), provocará los mayores riesgos en el sistema sino se maneja correctamente.

- **Requisitos funcionales:** Los requisitos funcionales definen lo que el sistema debería de hacer.

2.3.4 ANÁLISIS Y DISEÑO

Durante esta etapa es modelado el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos. Esto contribuye a una arquitectura sólida y estable que se convierte en un plano para la implementación. Los modelos desarrollados en esta etapa son más formales y específicos de una implementación. Durante esta etapa son desarrollados el Documento de Arquitectura, Diagramas de Clases, Diagramas de Entidad Relación, Diagrama de Despliegue entre otros.

ELEMENTOS CRÍTICOS DE LA ETAPA

- **Documento de la Arquitectura:**
    - Casos de uso arquitectónicamente significativos: Son aquellos que representan las partes más críticas de la arquitectura del sistema y demuestran la funcionalidad del sistema.

- **Modelo del Sistema:** Establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.
    - **Descripción de casos de uso asociados a la seguridad:** Son los casos de uso asociados a la seguridad del sistema.

2.3.5 IMPLEMENTACIÓN

En la implementación, a partir de los resultados abordados en las etapas anteriores y después de un análisis y diseño riguroso, se construye el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares.

ELEMENTOS CRÍTICOS DE LA ETAPA

- **Documento de la Arquitectura:**
    - El acceso a datos: Se refiere a la manera en que el arquitecto de base de datos define cómo se realizarán las consultas a la base de datos, teniendo en cuenta los requisitos de seguridad descritos en la etapa de requisitos.
    - **Herramientas verticales:**
        - **Frameworks**: Estructuras conceptuales y tecnológicas de soporte definido, normalmente con artefactos o módulos de software concretos, en

base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

- **Entorno seguro de desarrollo**: Se establece una infraestructura que dé soporte a los servicios de seguridad que se quieren proporcionar. Lo primero que hay que establecer es qué aplicaciones necesitan seguridad y cuántos servicios se necesitan. En segundo lugar hay que determinar cómo se van a proporcionar esos servicios, si van a ser transparentes al usuario, si se le va a dejar elegir el tipo de servicio, etc.

2.3.6 PRUEBAS

Son los procesos que permiten verificar y revelar la calidad de un producto software. Básicamente es una etapa en el desarrollo de software consistente en probar las aplicaciones construidas. En el ciclo de vida del programa de mejora las pruebas están comprendidas en dos etapas, una de los proyectos y otra de un ente externo al proyecto pero al final ambas utilizan los mismos mecanismos para llevar a cabo el proceso.

PRUEBAS INTERNAS

Durante esta etapa el proyecto verifica el resultado de la implementación probando según sea necesaria cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Durante esta etapa se deben desarrollar artefactos de prueba como: Diseños de casos de prueba, listas de chequeo y de ser posibles componentes de prueba ejecutables para automatizar las pruebas.

PRUEBAS DE LIBERACIÓN

Pruebas diseñadas e implementadas por el Laboratorio Industrial de Pruebas de Software a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

ELEMENTOS CRÍTICOS DE LA ETAPA

- **Los métodos de evaluación arquitectónicos:** Están basados principalmente en la evaluación de los atributos o cualidades arquitectónicas que responden a los requisitos no funcionales del software. El propósito de realizar evaluaciones a la arquitectura, es analizar e identificar riesgos potenciales en su estructura y propiedades que puedan afectar al sistema de software resultante, verificar que los requisitos no funcionales es-

tén presentes en la arquitectura, así como determinar en qué grado se satisfacen los atributos de calidad.

- Pruebas de seguridad: Tienen como principal objetivo verificar, antes de la liberación del sistema, la aplicación de los mecanismos de protección incorporados. Se realizan para detectar la existencia de vulnerabilidades y defectos de seguridad, para eliminarlos y con ello evitar el riesgo y costo que se ocasionaría en el ambiente final de operación del sistema.
  - Pruebas de comprobación del sistema de autenticación.
  - Pruebas de gestión de sesiones.
  - Pruebas de validación de datos.
  - Pruebas de gestión de configuración de la infraestructura.
  - Pruebas de autorización.
  - Pruebas de stress y carga.

2.3.7 DESPLIEGUE

Durante esta etapa se procede a la entrega de la solución, así como a la instalación, configuración, prueba y puesta en marcha del software en el entorno real del cliente. Las pruebas de esta etapa incluyen pruebas de aceptación y pruebas pilotos. También deben realizarse en este período la capacitación y acompañamiento a clientes para asegurar que adquieran los conocimientos necesarios en la manipulación del software.

ELEMENTO CRÍTICO DE LA ETAPA

- Los escenarios de despliegue: Son aquellos donde se va a utilizar el software en cuestión y deben estar en correspondencia con los que se definieron en los requisitos no funcionales de seguridad.

2.3.8 SOPORTE

Durante esta etapa, por un tiempo limitado, el proyecto ofrecerá un servicio para resolver conflictos y problemas de usabilidad y rendimiento del software entregado al cliente, suministrándole actualizaciones y parches a errores.

ELEMENTO CRÍTICO DE LA ETAPA

- **Las modificaciones al software:** Se efectuarán en la menor medida posible, deben estar en el rango de los acuerdos con los clientes, nunca más allá, ya que pueden afectar el código que no está implicado en estas modificaciones.

A continuación se muestra una tabla que recoge los elementos que se deben tener en cuenta durante todo el ciclo de desarrollo de software, en este caso solo las etapas en las que se puede influir en la seguridad del producto.

Etapa	Elementos Críticos

Modelación del Negocio	• Modelo de casos de uso del negocio.
	• Reglas del negocio.
	• Documento de la Arquitectura:
	  ○ Herramientas horizontales:
	  1. Repositorios.
	  2. Control de versiones.

Requisitos	• Requisitos no funcionales: ○ Requisitos de seguridad. • Requisitos funcionales.

Análisis y Diseño	• Documento de la Arquitectura:
	  ○ Casos de uso arquitectónicamente significativos.
	• Modelo del sistema:
	  ○ Descripción de casos de uso asociados a la seguridad.

Implementación	• Documento de la Arquitectura:
	  ○ El Acceso a datos.
	  ○ Herramientas verticales.
	  ▪ Frameworks.
	  ○ Entorno seguro de desarrollo.

Pruebas	• Pruebas de seguridad:
	  ○ Pruebas de comprobación del sistema de autenticación.
	  ○ Pruebas de gestión de sesiones.
	  ○ Pruebas de validación de datos.
	  ○ Pruebas de gestión de configuración de la infraestructura.
	  ○ Pruebas de autorización.
	  ○ Pruebas de stress y carga.

Despliegue	• Los escenarios de despliegue.
Soporte	• Modificaciones al software.

Tabla 1. Etapas y elementos críticos en cuanto al aseguramiento de la seguridad lógica en los sistemas informáticos de gobierno electrónico.

2.4 GUÍA DE ACTIVIDADES A REALIZAR POR ETAPAS

Etapa	Actividades
Modelación del Negocio	1. Identificar correctamente los roles y responsabilidades a través del modelado de casos de uso del negocio. 2. Especificar coherentemente las restricciones y derivaciones del negocio. 3. Establecer correctamente las herramientas horizontales

	en el Documento de la Arquitectura, como los repositorios y el control de versiones del proyecto.
Requisitos	1. Seleccionar la tecnología de software adecuada: • Lenguajes de programación. • Gestores de bases de datos. • Servidores de aplicaciones. • IDE de desarrollo. • Sistemas operativos. • Frameworks. • Antivirus. 2. Identificar y describir correctamente los requisitos funcionales de seguridad: • Enmascarar datos confidenciales cuando se visualicen (por ejemplo, contraseñas, cuentas). • Bloquear la cuenta del usuario después de cierto número de intentos de acceso fallidos. • No mostrar al usuario errores específicos de validación como resultado de un acceso fallido. (Darle ejemplos a los probadores de cómo crear esto). • Solamente permitir contraseñas alfanuméricas, que incluyan caracteres especiales y que tengan seis caracteres mínimos de longitud, todo esto para limitar ataques desde la interfaz. • Mostrar mensajes genéricos de error en la validación de credenciales para mitigar los riesgos de cosecha/enumeración de cuentas de usuario. • Permitir la funcionalidad de cambio de contraseña únicamente a usuarios autenticados validando la antigua contraseña, la nueva contraseña y la respuesta a la pregunta de seguridad, esto para evitar ataques de fuerza bruta a la contraseña a través de la funcionalidad de cambio de contraseña.

	• Cambiar de manera periódica las contraseñas para los usuarios autorizados. • Cifrar contraseñas. • Autenticación. • Gestión de roles.
Análisis y Diseño	1. Determinar correctamente en el Documento de la Arquitectura los casos de uso arquitectónicamente significativos, los cuales representan las partes más críticas de la arquitectura del sistema y demuestran la funcionalidad del sistema. 2. Analizar y describir correctamente los casos de uso asociados a la seguridad, ya que estos podrían afectar negativamente el desempeño del sistema.
Implementación	1. En el Documento de la Arquitectura tener un extremo cuidado con el acceso a datos: • Reducir la cantidad de accesos a los datos. • Mostrar solamente la información necesaria. 2. Tener en cuenta una adecuada selección de las herramientas verticales como el o los frameworks que se van a emplear para el desarrollo de la aplicación. 3. Asegurar un entorno seguro de desarrollo en el que se va a implementar el software, donde pueda estar extinto de riesgos y amenazas para su seguridad, durante las etapas de construcción. 4. El desarrollador deberá tener en cuenta la incidencia de las vulnerabilidades detectadas hasta el momento para evitar incurrir en estas brechas de seguridad: • Revelación de información a través del banner. • Autocompletamiento habilitado. • Validación de código.

Implementación	• Gestión de sesiones. • Brechas del phpMyAdmin. • Mostrar la contraseña en texto plano. • Cross-site Scripting. • Permisos para adjuntar archivos ejecutables. • Pykto plugin (Vulnerabilidades de las URLs). • Path Disclosure. 5. El desarrollador deberá seguir algunas instrucciones para escribir código seguro: • Utilizar herramientas de análisis de código (estas herramientas encuentran errores ocultos con mayor eficacia y menos esfuerzo). • Validar todos los datos introducidos por los usuarios. • Evitar las saturaciones de búfer. • Reutilizar código que se sabe seguro. • Tener cuidado con los valores suministrados por el usuario. • Tener cuidado con el manejo de ficheros. • Proteger las sesiones. • Utilizar la criptografía. • Tener en cuenta los estándares de codificación.
Prueba	1. Analizar e identificar riesgos potenciales en cuanto a estructura y propiedades del modelo arquitectónico. 2. Realizar las diferentes pruebas de seguridad correctamente para comprobar si el software que se desarrolla es seguro: • Pruebas de comprobación del sistema de autenticación.

	• Pruebas de gestión de sesiones. • Pruebas de validación de datos. • Pruebas de gestión de configuración de la infra-estructura. • Pruebas de autorización. • Pruebas de stress y carga.
Despliegue	1. Verificar correspondencia del escenario de despliegue con los requisitos no funcionales.
Soporte	1. Reducir la cantidad de modificaciones al software.

Tabla 2. Etapas y actividades para el aseguramiento de la seguridad lógica en los sistemas informáticos de gobierno electrónico.

2.5 CONCLUSIONES PARCIALES

En este capítulo se realizó un estudio del ciclo de vida del proceso de desarrollo de software basado en el programa de mejoras, en el cual se encuentra la Universidad de las Ciencias Informáticas. De cada etapa establecida por este se determinaron los elementos críticos con respecto a la seguridad y su afectación a la calidad de software. Se propone un guía en aras de asegurar la seguridad y de esta forma contribuir a calidad de software.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

3.1. INTRODUCCIÓN

Con la finalidad de validar la propuesta expuesta en el capítulo anterior se realizó una encuesta a un grupo de expertos para conocer su criterio sobre la aplicación de la investigación y los resultados que puede traer a la universidad. Se empleó el Método Delphi para la validación, por ser este un método sencillo, rápido y eficaz. Tiene como objeto, la recopilación de opiniones de expertos sobre un tema particular, con el fin de incorporar dichos juicios en la configuración de un cuestionario y conseguir un consenso a través de la convergencia de las opiniones de los mismos para así validar la propuesta de solución. También se caracteriza por su habilidad para guiar la opinión del grupo hacia una decisión final, sin forzar falsos consensos. El empleo de este método permitió conocer la opinión y valoración de los expertos seleccionados. En aras de determinar el grado de concordancia entre los expertos, con respecto a las evaluaciones se utilizó el coeficiente de Kendall de la herramienta Paquete Estadístico para las Ciencias Sociales (SPSS).

3.2. MÉTODO DELPHI.

El método Delphi, ideado a comienzos de los años 60 en el Centro de Investigación estadounidense RAND Corporation por Olaf Helmer y Theodore J. Gordon, como un instrumento para realizar predicciones. Consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes. (Astigarraga, 2004)

3.2.1. PRINCIPALES CARACTERÍSTICAS DEL MÉTODO DELPHI

1. **Anonimato**: Los expertos contestan las preguntas sin consultarse mutuamente (por lo que es recomendable que dos expertos no conozcan entre sí que están opinando sobre un mismo tema).
2. **Retroalimentación controlada**: Después de cada ronda de preguntas, se tabulan las respuestas y se procesan antes de la siguiente ronda, para que los participantes puedan evaluar los resultados de la ronda anterior, así como las razones dadas para cada respuesta y su dispersión del promedio (esto permite que aumente el acuerdo al transcurrir varias rondas del proceso).

3. **Respuesta estadística del grupo:** El procesamiento de cada ronda se realiza con métodos estadísticos. Esto es la característica más importante que diferencia a este método de otros subjetivos. (Almaguer, 2006)

3.2.2. ELECCIÓN DE LOS EXPERTOS

Los expertos dispuestos deberán evaluar la propuesta individualmente y de forma anónima. Se trata de llegar a un consenso y analizar los aspectos de discrepancia, permitiendo además que:

- Ningún miembro del grupo de expertos sea influenciado por la reputación de otro de los miembros.
- Un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
- El experto pueda defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.

Se debe determinar el área del conocimiento que dominan los expertos, para así establecer la población indicada para evaluar la propuesta y selecciona la muestra a la cual se le aplicará el cuestionario.

- El área del conocimiento: Seguridad lógica.
- Población seleccionada: Arquitectos de software, líderes de proyecto, asesores de calidad y analistas principales de proyectos de CEGEL.
- Muestra: 5 individuos.

Para la selección de los expertos se escogieron, teniendo en consideración que cumplieran con estos requisitos:

- Desempeñan o se han desempeñado en el rol de asesor de calidad en proyectos o en los centros de la facultad 3.
- Desempeñan o se han desempeñado como jefes de departamento, de asignaturas o miembros del departamento de Ingeniería de Software en la facultad 3 o a nivel central.
- Desempeñan o se han desempeñado como arquitecto o analista principal de los proyectos de la facultad 3.

- Desempeñan o se han desempeñado como profesores de la asignatura Seguridad Informática.
- Pertenecen o colaboran con grupos investigativos de seguridad informática o calidad de software.
- Se encuentran en el proceso de maestría o doctorado en los temas de seguridad informática o calidad de software.
- Desempeñan el rol de líder de proyectos de calidad del CEGEL.

De los expertos, dos han trabajado directamente en proyectos de convenio con Venezuela cumpliendo misión en ese país, los otros tres en proyectos nacionales de gran envergadura para el gobierno electrónico, todos tienen más de tres años de experiencia como jefes de proyecto y arquitectos de software.

Para determinar el coeficiente de experticia de los expertos se aplicó una encuesta de autovaloración. (Ver anexo 2).

El coeficiente de experticia (K): Se determina por la opinión del encuestado sobre su nivel de conocimiento respecto al área del conocimiento en cuestión u objeto de indagación. Este no se asocia al grado científico, tarea, labor o responsabilidad que desempeña. Su formulación se describe a continuación.

$$K = 1/2(Kc + Ka)$$

Si 0,8 ≤ K < 1,0 el coeficiente de experticia es alto.

Si 0,5 ≤ K < 0,8 el coeficiente de experticia es medio.

Si K < 0,5 el coeficiente de experticia es bajo.

El coeficiente de conocimiento (Kc): Es el conocimiento o información que tiene el experto acerca del área de conocimiento. Sus valores se encuentran representados en una escala del 0 al 10, (Ver anexo 2) que se obtienen por la autovaloración del experto. Estos valores se multiplican por 0,1 obteniéndose así el coeficiente de conocimiento. El "0" indica desconocimiento del área de conocimiento, mientras que el "10" significa pleno conocimiento de la misma.

De acuerdo al cuestionario aplicado, se obtuvo el siguiente resultado con respecto a la autovaloración de los expertos.

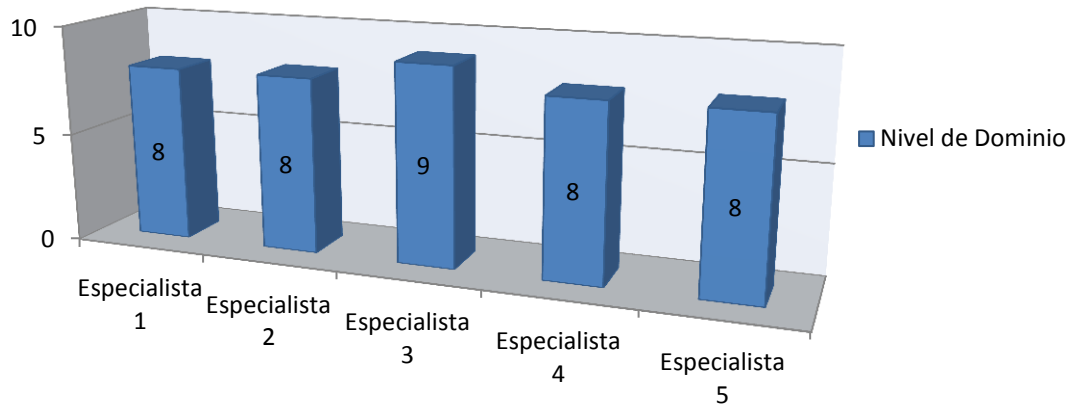


Figura 9. Valores de la autovaloración de los expertos para calcular sus coeficientes de conocimiento.

El coeficiente de argumentación (Ka): Es la argumentación o fundamentación de los criterios del experto. Se obtiene como resultado de la suma de los puntos alcanzados a partir de la tabla patrón que se muestra a continuación:

Fuentes de argumentación	**Grado de influencia de cada una de las fuentes en sus criterios.**		
	Alto	**Medio**	**Bajo**
Análisis teóricos realizados	0.3	0.2	0.1
Experiencia obtenida	0.5	0.4	0.2
Trabajos de autores extranjeros consultados	0.2	0.15	0.1

Tabla 3. Grado de influencia de cada una de las fuentes en sus criterios.

Por medio de la autovaloración de los expertos se pudo conocer algunos aspectos fundamentales, como son los conocimientos teóricos que poseen acerca del tema, la experiencia obtenida en la actividad práctica y trabajos de autores extranjeros consultados sobre el tema y así determinar los coeficientes de argumentación.

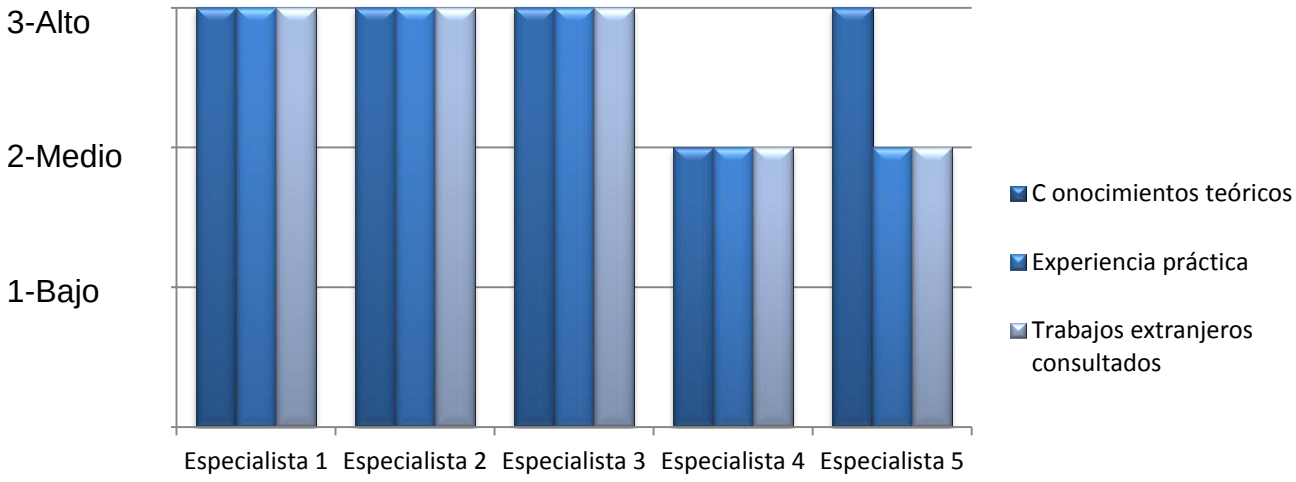


Figura 10. Valores de la autovaloración de los expertos para calcular sus coeficientes de argumentación.

Experto	Kc	Ka	K	Nivel
1	0,8	1	0,9	**Alto**
2	0,8	1	0,9	**Alto**
3	0,9	1	0,95	**Alto**
4	0,8	0,75	0,775	**Medio**
5	0,8	0,85	0,825	**Alto**

Tabla 4. Coeficiente de experticia.

Figura 11. Coeficiente de experticia.

En la gráfica se muestra los coeficientes de experticia de cada experto, de lo que se puede calcular que el promedio es de 0,87. Lo anteriormente planteado demuestra un alto nivel en los expertos, lo que posibilita una correcta validación de la propuesta de solución.

3.3. RESULTADOS DE LA APLICACIÓN DEL CUESTIONARIO

Se analizaron los resultados de la aplicación del cuestionario, donde se evalúa, por parte de los expertos, cada elemento definido en la solución.

En el cuestionario se formulan siete preguntas dirigidas a valorar los elementos críticos definidos para asegurar la característica de calidad de software, durante las etapas del ciclo de desarrollo de software, basado en el programa de mejora en el que se encuentra la UCI.

Las respuestas a las preguntas han sido plasmadas en la siguiente tabla, de manera que se puedan comprender mejor los resultados:

Preguntas	Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	No Adecuado
Pregunta 1 Modelo Negocio	2	3			
Pregunta 2 Requisitos	2	2	1		
Pregunta 3 Análisis Diseño	1	2	2		
Pregunta 4 Implementación	4	1			
Pregunta 5 Prueba	3	2			
Pregunta6 Despliegue	2	3			
Pregunta 7 Soporte		2	3		

Tabla 5. Evaluación por preguntas de los expertos.

El proceso de evaluación es iterativo, se realizan varias rondas hasta que haya un consenso entre los expertos, en este caso solo se realizó una ronda. El cuestionario se le envió a cada uno por separado, para no condicionar las respuestas de los demás.

A continuación se muestra toda la información recopilada a partir de la aplicación del instrumento.

En la respuesta a la **primera pregunta:**

> ¿Considera usted que el modelo de Casos de Uso del Negocio, las reglas del Negocio y las herramientas horizontales como repositorios y control de versiones en el Documento de Arquitectura influyen en la seguridad de los sistemas?

Los expertos encuestados *respondieron lo siguiente:*

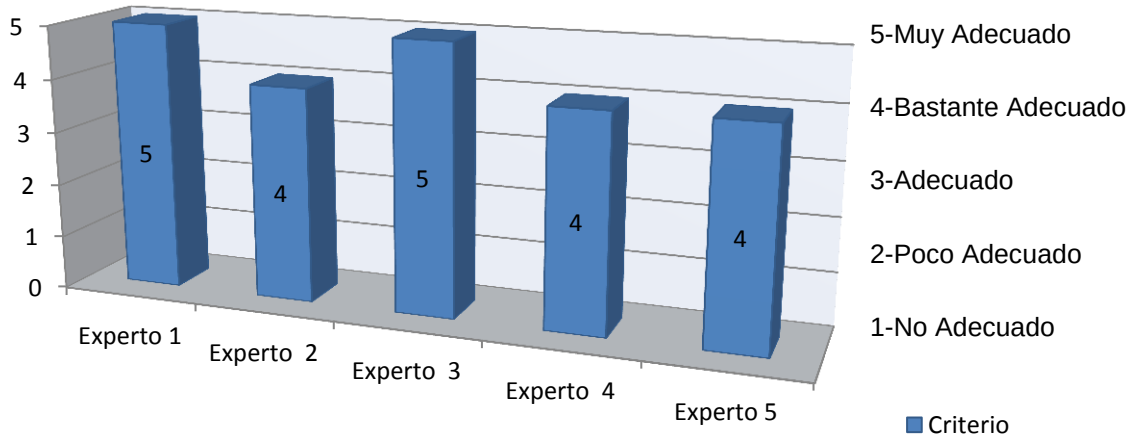


Figura 12. Criterio de evaluación de la pregunta No.1 por expertos.

El 100 % de los expertos valoró entre Adecuado y Muy Adecuado las respuestas a la pregunta No.1, como se evidencia a en la gráfica siguiente.

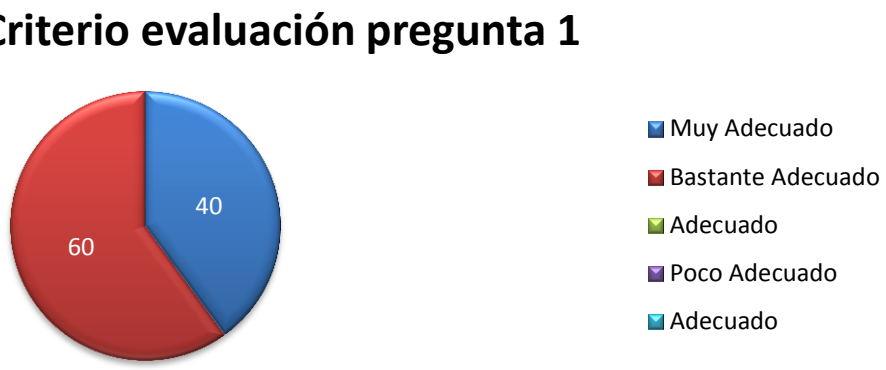


Figura 13. Por ciento del criterio de evaluación de los expertos para la pregunta No.1.

En respuesta a la **segunda pregunta:**

> ¿Considera usted que definir correctamente los requisitos no funcionales como: requisitos de seguridad y definición correcta del software, así como la descripción correcta de los requisitos funcionales del sistema que tributen a la seguridad influyen en la seguridad del mismo?

Los expertos encuestados respondieron lo siguiente:

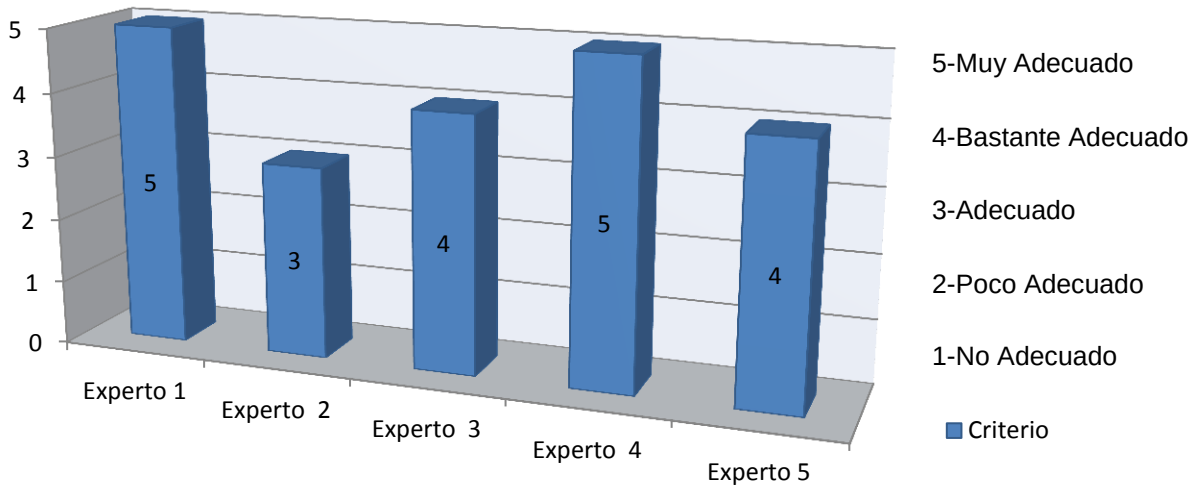


Figura 14. Criterio de evaluación de la pregunta No.2 por expertos.

El 100 % de los expertos valoró entre Adecuado y Muy Adecuado los elementos planteados en la pregunta No. 2 como se aprecia a continuación.

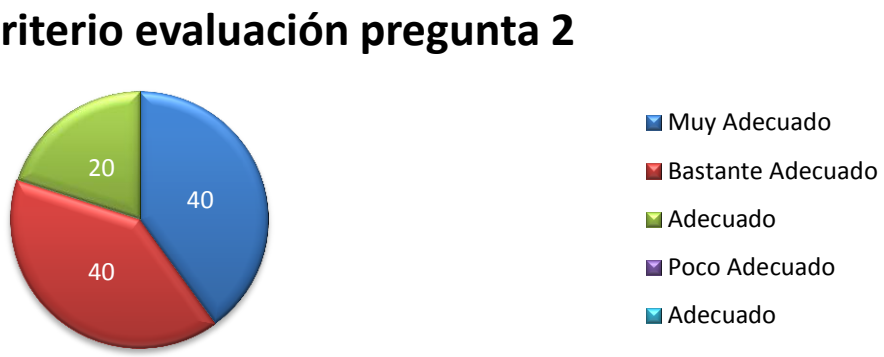


Figura 15. Por ciento del criterio de evaluación de los expertos para la pregunta No.2.

En respuesta a la **tercera pregunta:**

> ¿Considera usted que desarrollar correctamente los casos de uso arquitectónicamente significativos, analizar los casos de uso que tengan fuertes vínculos con la seguridad y realizar una correcta descripción de casos de uso asociados a la seguridad, son los elementos primordiales en el aseguramiento de la seguridad durante la etapa de Análisis y Diseño?

Los expertos encuestados respondieron lo siguiente:

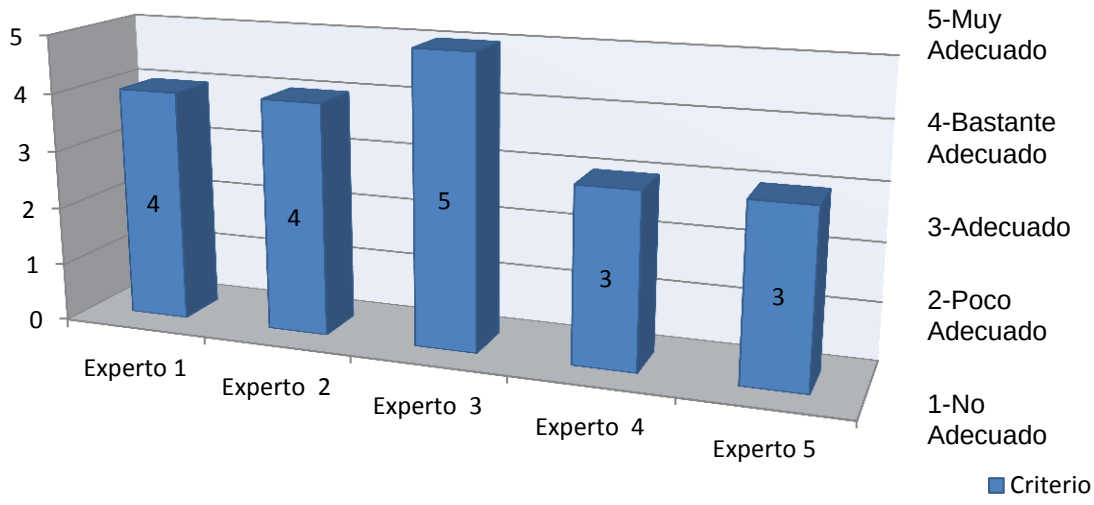


Figura 16. Criterio de evaluación de la pregunta No.3 por expertos.

El 100 % de los expertos valoró entre Adecuado y Muy Adecuado los elementos planteados en la pregunta No. 3, como se aprecia a continuación.

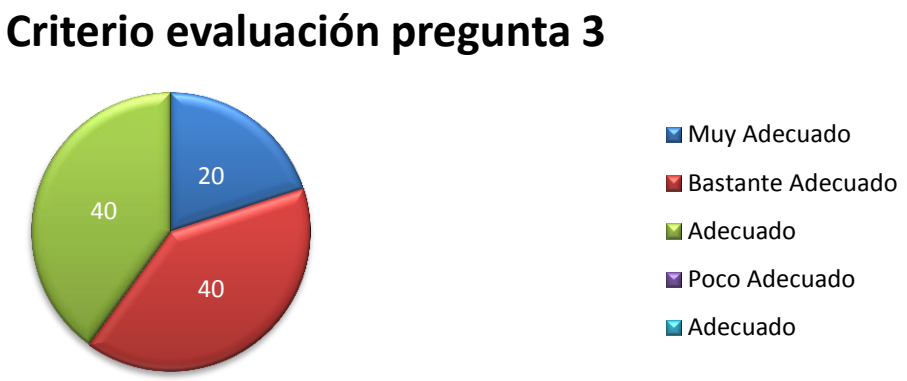


Figura 17. Por ciento del criterio de evaluación de los expertos para la pregunta No.3.

En respuesta a la **cuarta pregunta**:

> ¿Considera usted que el acceso a datos, la adecuada selección de las herramientas verticales como los frameworks y asegurar un entorno seguro de desarrollo en el que se va a desarrollar el software, son los elementos críticos que pueden influir en la seguridad del sistema?

Los expertos encuestados respondieron lo siguiente:

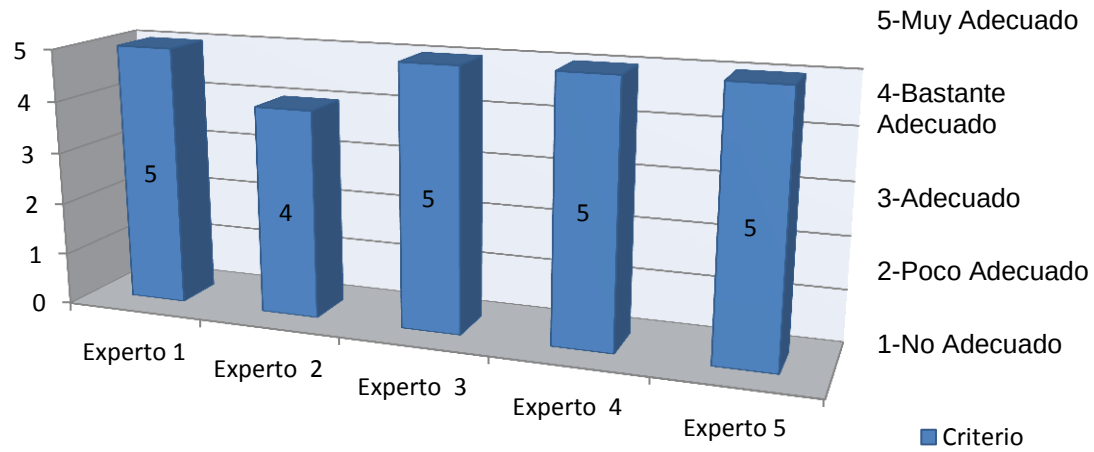


Figura 18. Criterio de evaluación de la pregunta No.4 por expertos.

El 100 % de los expertos valoran entre Bastante Adecuado y Muy Adecuado los elementos planteados en la pregunta No. 4, como se aprecia a continuación.

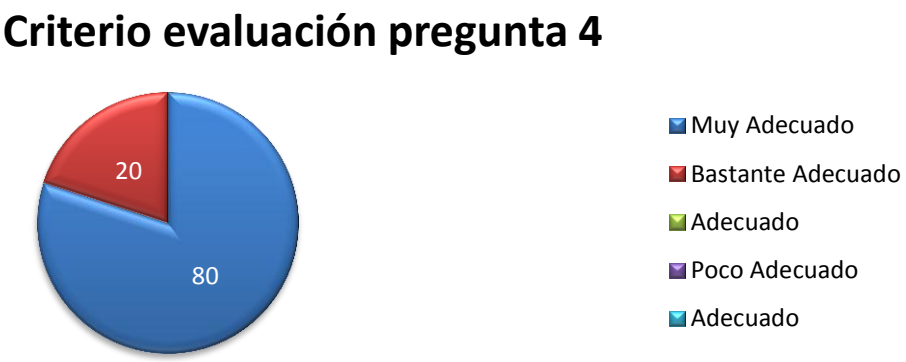


Figura 19. Por ciento del criterio de evaluación de los expertos para la pregunta No.4.

En respuesta a la **quinta pregunta:**

> ¿Considera usted que la realización de las diferentes pruebas de seguridad como: pruebas de comprobación del sistema de autenticación, pruebas de gestión de sesiones, pruebas de validación de datos, pruebas de gestión de configuración de la infraestructura, pruebas de autorización, pruebas de stress y de carga, puedan influir en la seguridad del sistema?

Los expertos encuestados respondieron lo siguiente:

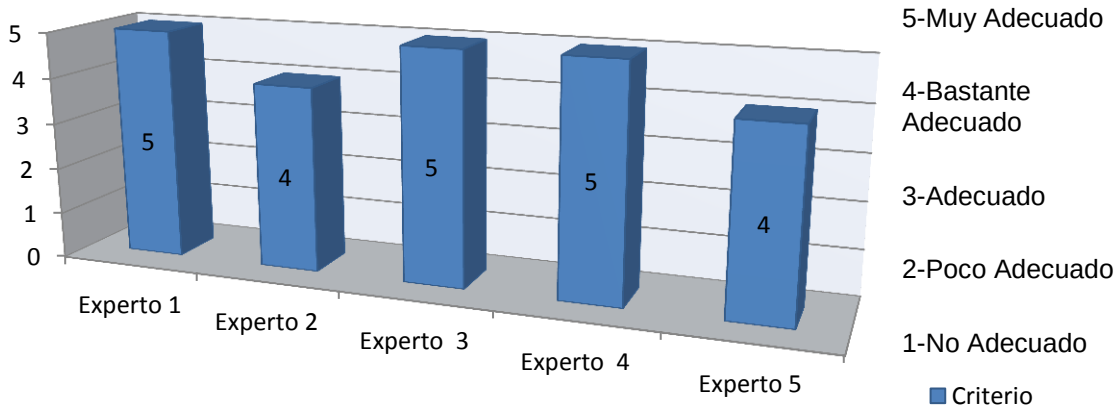


Figura 20. Criterio de evaluación de la pregunta No.5 por expertos.

El 100 % de los expertos valoran entre Bastante Adecuado y Muy Adecuado los elementos planteados en la pregunta No. 5, como se aprecia a continuación.

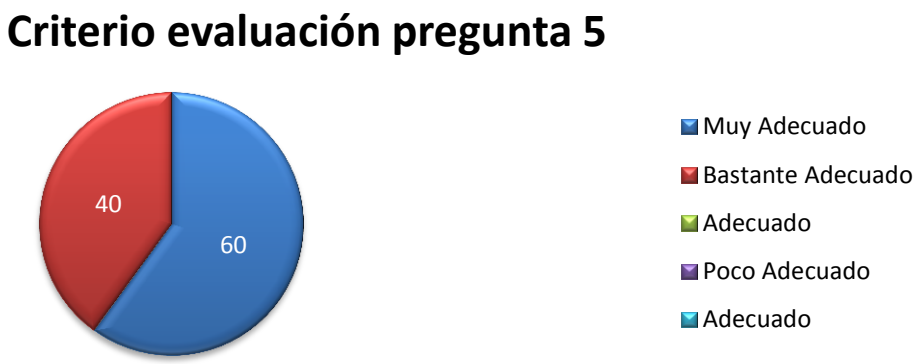


Figura 21. Por ciento del criterio de evaluación de los expertos para la pregunta No.5.

En la **sexta pregunta**

> ¿Considera usted que la correcta definición de los escenarios de despliegue durante la etapa de Despliegue es un elemento crítico para asegurar la seguridad?

Los expertos encuestados respondieron lo siguiente:

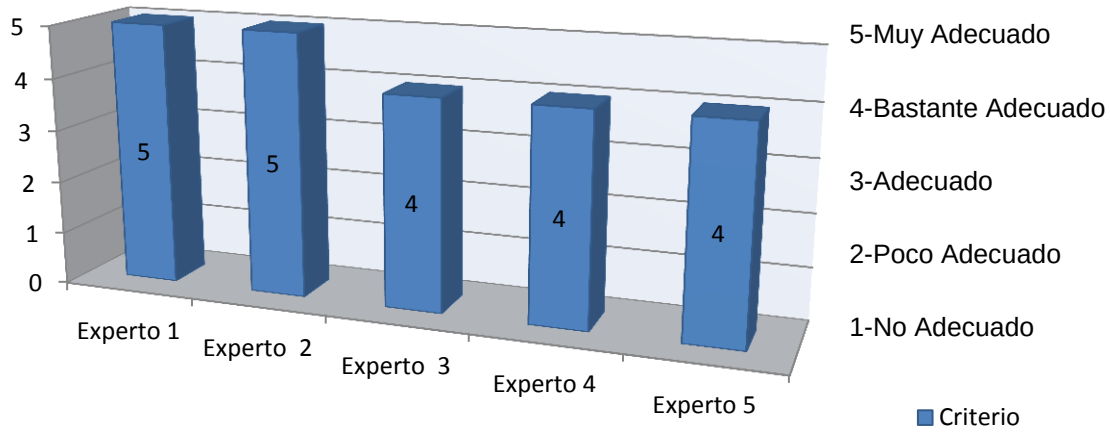


Figura 22. Criterio de evaluación de la pregunta No.6 por expertos.

El 100 % de los expertos valora entre Adecuado y Muy Adecuado los elementos planteados en la pregunta No.6, como se aprecia a continuación.

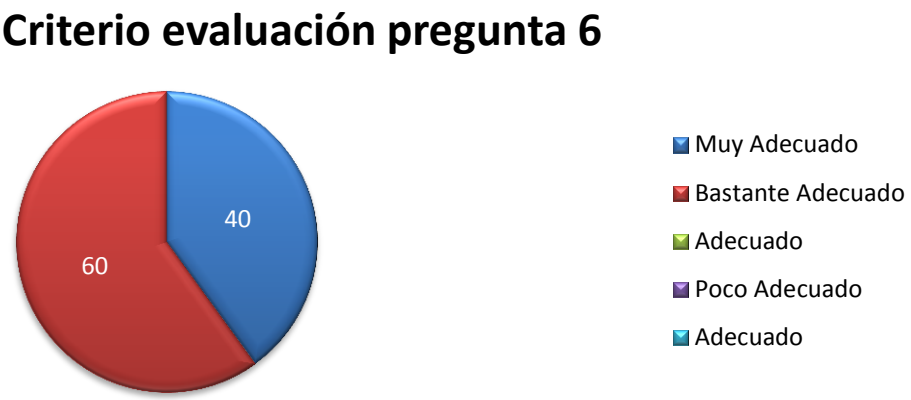


Figura 23. Por ciento del criterio de evaluación de los expertos para la pregunta No.6.

En respuesta a la **séptima pregunta**

> ¿Considera usted que la cantidad de modificaciones al software en la etapa de Soporte influya en la seguridad del sistema?

Los expertos evaluaron lo siguiente:

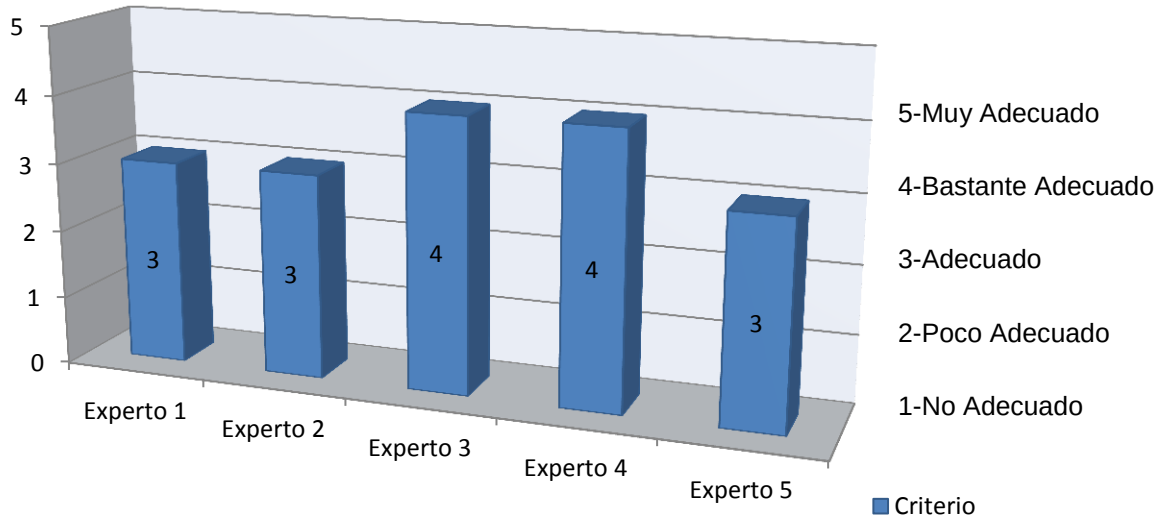


Figura 24. Criterio de evaluación de la pregunta No.7 por expertos.

El 100 % de los expertos valoró entre Adecuado y Muy Adecuado los elementos planteados en la pregunta No. 7, como se aprecia a continuación.

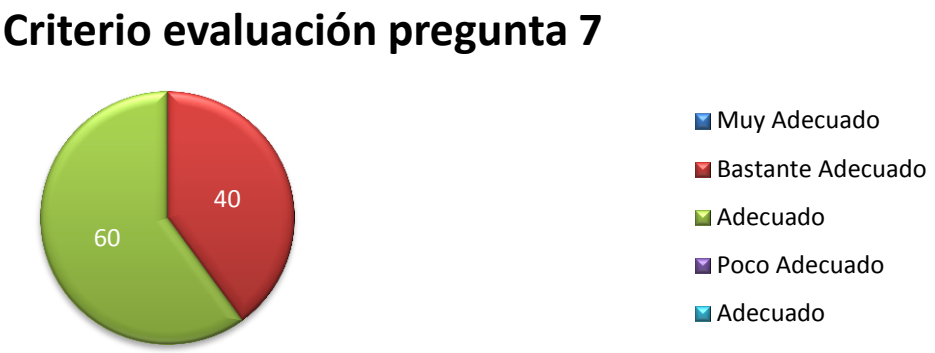


Figura 25. Por ciento del criterio de evaluación de los expertos para la pregunta No.7.

3.4. COEFICIENTE DE KENDALL

Se pudo determinar cómo los expertos opinan que la guía propuesta tiene un valor significativo para la seguridad de los sistemas. Los resultados arrojados por la encuesta a los expertos fueron satisfactorios, pero para mayor confirmación se analizan los resultados estadísticamente, por lo que se procede a determinar el grado de concordancia entre los expertos, con respecto a las evaluaciones que hicieron. Para ello se determinó utilizar el coeficiente de Kendall (W) para la explotación de los datos, obteniéndose como resultado:

Ranks

	Mean Rank
p1	4,50
p2	3,80
p3	2,50
p4	5,70
p5	5,20
p6	4,50
p7	1,80

Test Statistics

N	5
Kendall's W[a]	,672
Chi-Square	20,157
df	6
Asymp. Sig.	,003

a. Kendall's Coefficient of Concordance

Figura 25. Coeficiente de Kendall.

Esta tabla muestra el número de casos válidos (N), el valor del estadístico W (W de Kendall), su transformación en Chi-cuadrado (Chi-Square), sus grados de libertad (df) y el nivel crítico (Asymp. Sig.). Para saber si W de Kendall es significativamente distinta de 0 se realizó una prueba de hipótesis donde:

H_0 = No hay concordancia entre los expertos.

H_1 = Hay concordancia entre los expertos.

El coeficiente W de Kendall es una medida de la concordancia de los expertos y por definición del Método Delphi, el resultado debe moverse en un rango de 0 a 1 y debe ser siempre W > 0,5 porque cuanto más se acerque el coeficiente a 1, mayor será el grado de concordancia entre los expertos. En el estudio realizado resultó ser un aproximado de W= 0,672, por tanto la propuesta resultó ser aceptada y con un nivel de concordancia alto con respecto a los criterios que fueron evaluados.

La tabla muestra que el valor del nivel crítico (0,003) es menor que 0,05, por lo que se puede rechazar la hipótesis nula y concluir que existe asociación significativa en las preguntas realizadas a los expertos.

Después de efectuada la primera ronda se confirmó que no era necesario realizar una segunda ronda, pues existía un nivel de concordancia entre los expertos evaluando los elementos seleccionados entre Muy Adecuados y Adecuados.

3.5. CONCLUSIONES PARCIALES

La utilización de este instrumento de validación fue positiva, ya que se pudo determinar cómo los expertos, que poseen un alto nivel de experticia, opinan que la estrategia propuesta tiene un objetivo que se corresponde con la necesidad creciente en las nuevas tecnologías.

Las respuestas de la encuesta a los expertos, para la validación de la propuesta de solución fueron:

- El 40% Muy Adecuada.
- El 42.86% Bastante Adecuada.
- El 17.14% Adecuada.

El 100% de los criterios se hallaron desde Adecuado hasta Muy Adecuado, por cuanto permitió corroborar que los elementos definidos en la propuesta pueden influir positivamente en la seguridad del sistema. La estrategia quedó satisfactoriamente validada demostrándose así que esta guía contribuirá con su aplicación a la reducción de vulnerabilidades en los sistemas informáticos de gobierno electrónico.

CONCLUSIONES

Con la realización de este trabajo de diploma se analizaron conceptos de vital importancia para el tema de esta investigación como son: calidad de software, aseguramiento de la calidad, modelos y estándares de calidad de software, seguridad y seguridad lógica. Se seleccionó satisfactoriamente el modelo de calidad que sirvió como guía para el desarrollo de la propuesta de solución.

Se definió, por parte de los autores, elementos críticos por cada etapa del ciclo de vida del proceso de desarrollo de software que plantea el programa de mejora. A partir de los elementos críticos, se desarrolló una guía de actividades, teniendo en cuenta cada una de las etapas y su contribución a la seguridad lógica de los sistemas informáticos de gobierno electrónico.

La propuesta de solución fue validada satisfactoriamente, con un 100% de los criterios de los especialistas entre Adecuado y Muy Adecuado. Por tanto, se hace necesaria la aplicación de esta guía, para obtener un grado de certificación que le permita a los productos obtenidos, compararse con los de más alto nivel de aceptación y comercialización internacionalmente.

RECOMENDACIONES

- Aplicar la guía propuesta en otros centros.

- Actualizar la guía de forma periódica, de acuerdo a lo que se defina en cada centro o departamento.

- Continuar realizando estudios sobre la seguridad en general, teniendo en cuenta el constante cambio de protocolos, herramientas y tecnologías usadas para el desarrollo de aplicaciones informáticas y el surgimiento de nuevas amenazas.

- Aplicar la guía durante todo el ciclo de desarrollo de los proyectos y comparar los resultados obtenidos, para saber si ha mejorado o no la seguridad de sus productos con la aplicación de esta guía.

BIBLIOGRAFÍA

Almaguer, González. Armín. 2006. *El método Delphi y el procesamiento estadístico de los datos obtenidos de la consulta a los expertos.* Ciudad de la Habana, Cuba : ISP “José de la Luz y Caballero”, 2006.

Astigarraga, Eneko. 2004. *EL MÉTODO DELPHI.* Donostia, San Sebastian, España : Universidad de Deusto, Facultad de CC.EE. y Empresariales, ESTE, 2004.

Bertone, Rodolfo. Thomas, Pablo. 2010. *Planificación Estratégica de Proyectos de Software CACIC.* La Plata, Buenos Aires, Argentina : UNLP, Universidad Nacional de La Plata, Facultad de Informática, 2010.

Beth Chrissis, Mary. Konrad,Mike .Shrum,Sandy. 2009. *Guía para la integración de procesos.* Madrid,España : Pearson Educación, 2009.

blogspot. 2010. Seguridad Informática. [En línea] 2010. http://seguridadinformaica.blogspot.com/p/tecnicas-para-asegurar-un-sistema.html.

Camacho, Erika. Cardeso, Fabio. Nuñes, Gabriel. 2004. *Arquitectura de software. Guía de estudio.* 2004.

Cardona, Ing. Diego. 2002. *“El gobierno electrónico Una revisión desde la perspectiva de la prestación de servicios”.* Barcelona, España : s.n., 2002.

Chan Basto, Lorenzo Alberto. Solís Sosa, Sergio Alberto. 2010. Tecnologías de la información. [En línea] Universidad de Quintana Roo, México, 2010. http://ti.uqroo.mx/htm/seguridad_logica.html.

Demand Media, Inc. 2012. eHow.com. [En línea] 2012. http://www.ehow.com/facts_5762097_iso-standards-definition.html.

Eickelman, N. and Hayes, J. 2004. *“New Year’s Resolutions for Software Quality”, IEEE Software.* 2004.

Espino, Raquel Espino. 2011. *La Calidad: Una oportunidad.* Islas Canarias, España : ULPGC, Universidad de Las Palmas de Gran Canarias, 2011.

García, Óscar Gómez. 2009. *SQuaRE: Una unificación de normas para la especificación de requisitos y la evaluación de la calidad.* Castilla-La Mancha,España : Universidad de Castilla-La Mancha, 2009.

Hernandez, Yohamnes. 2009. *Seguridad en proyectos.* La Habana : s.n., 2009.

Huerta, Antonio Villalón. 2000. Seguridad en Unix y Redes. [En línea] 2000. http://www.kriptopolis.org.

Lang, Jean-Philippe. 2011. CEGEL (Centro de Gobierno Electrónico). [En línea] Paquete de Gestión de Proyectos (GESPRO 11.05). Laboratorio de Soluciones e Investigaciones Avanzadas en Gestión de Proyectos, UCI, 2011. http://portal.cegel.prod.uci.cu/.

Losavio, F. Chirinos, L. Lévy, N. & Ramdane-Cherif, A. 2003. *Quality Characteristics for Software Architecture.* s.l. : JOT, 2003.

Maña, Antonio. Ray, Diego. Sánchez, Francisco. I.Yagüe, Mariemma. 2004. *Integrando la Ingeniería de Seguridad en un Proceso de Ingeniería Software.* Málaga : Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, 2004.

Mario, Farías-Elinos, M. en C. 2002. [En línea] Universidad La Salle,México D.F., 2002. www.ulsa.edu.mx.

Mark G. Graff, Kenneth R. van Wyk. 2003. *Secure Coding: Principles & Practices.* Estados Unidos de América : O'Reilly, 2003.

Morilla, José Joaquín Ruiz. 2009. *Calidad y Medición de Sistemas de Información.* Sevilla, España : Universidad de Sevilla, 2009.

NIST. 2010. National Institute of Standars and Technology. [En línea] U.S. Department of Commerce., 2010. http://www.nist.gov.

Oyarzún L., Fernando. 2006. *Gestión de la calidad.* Iquique, Chile : UNAP, Universidad Arturo Prat, 2006.

Pressman, Roger S. 2002. *Ingeniería del Software Un Enfoque Práctico.5ta. Edición.* Madrid, España : McGraw-Hill / Interamericana de España, S. A. U., 2002.

Quiñones A., Ernesto. 2006. APESOL. *Asociación peruana de software libre.* [En línea] 2006. htt://www.apesol.org.pe.

RAE. 2010. Real Academia Española. [En línea] 2010. http://www.rae.es/rae.html.

Ramón Gil-García, Luis Felipe Luna Reyes. 2007. *Modelo multi-dimensional de medición del gobierno electrónico para América Latina y el Caribe.* Santiago de Chile, Chile : Naciones Unidas, 2007.

Scalone, Lic. Fernanda. 2006. *Maestría en Ingeniería en Calidad "Estudio comparativo de los modelos y estandares de calidad del software".* Buenos Aires, Argentina : Universidad Tecnológica Nacional Facultad Regional Buenos Aires, 2006.

ANEXOS

ANEXO 1

ENTREVISTA APLICADA A LOS PROYECTOS DE CEGEL PARA VERIFICAR CÓMO ASEGURAN LA SEGURIDAD LÓGICA EN SUS PRODUCTOS.

1. ¿Utilizan algún modelo o estándar para asegurar la seguridad en sus productos?

   _____

2. ¿Utilizan buenas prácticas de programación durante el ciclo de vida del proyecto?

   _____

3. ¿Realizan pruebas de seguridad en las distintas etapas de desarrollo del producto?

   _____

4. ¿Cuentan con especialistas de seguridad en el proyecto?

   _____

5. ¿Realizan pruebas de seguridad a aplicaciones de escritorio?

   _____

ANEXO 2

ENCUESTA APLICADA A ESPECIALISTAS DE CEGEL PARA SOMETER A SUS CRITERIOS LA PROPUESTA DE LA GUÍA PARA EL ASEGURAMIENTO DE LA SEGURIDAD EN LOS SISTEMAS INFORMÁTICOS DE GOBIERNO ELECTRÓNICO.

Compañero (a): Se desea someter a la valoración de un grupo de especialistas la propuesta de la Guía para el aseguramiento de la seguridad en los sistemas informáticos de gobierno electrónico. Con este fin se le solicita su valiosa colaboración, y sus opiniones se tendrán en cuenta para la elaboración de la Guía para el aseguramiento de la seguridad en los sistemas informáticos de gobierno electrónico.

Muchas Gracias.

1. Marque con una X el grado de dominio que usted posee en el tema de la investigación que se desarrolló (aseguramiento de la seguridad en sistemas informáticos de gobierno electrónico), considerando 0 como no tener ningún conocimiento y 10 el de pleno conocimiento del tema.

0	1	2	3	4	5	6	7	8	9	10

2. Valore el grado de influencia que cada una de las fuentes que se le presentan a continuación y si ha tenido en su conocimiento criterios sobre el tema que se investiga.

Fuentes de argumentación	Grado de influencia de cada una de las fuentes en sus criterios		
	A (Alto)	M (Medio)	B (Bajo)
Conocimientos teóricos que posee acerca del tema.			
Su experiencia sobre el tema obtenida en la actividad práctica.			
Trabajos de autores extranjeros consultados sobre el tema.			

3. Exprese su criterio acerca del elemento definido en la etapa de Modelación del Negocio.

Criterio de Especialistas					
Pregunta No. 1	**C1 Muy Adecuado**	**C2 Bastante Adecuado**	**C3 Adecuado**	**C4 Poco Adecuado**	**C5 No Adecuado**
¿Considera usted que el modelo de Casos de Uso del Negocio, las reglas del Negocio y las herramientas horizontales como repositorios y control de versiones en el Documento de Arquitectura influyen en la seguridad de los sistemas?					

4. Exprese su criterio acerca de los elementos definidos en la etapa de Requisitos.

Criterio de Especialistas					
Pregunta No. 2	**C1 Muy Adecuado**	**C2 Bastante Adecuado**	**C3 Adecuado**	**C4 Poco Adecuado**	**C5 No Adecuado**
¿Considera usted que definir correctamente los requisitos no funcionales como: requisitos de seguridad y definición correcta del software, así como la descripción correcta de los requisitos funcionales del sistema que tributen a la seguridad influyen en la seguridad del mismo?					

5. Exprese su criterio acerca de los elementos definidos en la etapa de Análisis y Diseño.

Criterio de Especialistas					
Pregunta No. 3	**C1 Muy Adecuado**	**C2 Bastante Adecuado**	**C3 Adecuado**	**C4 Poco Adecuado**	**C5 No Adecuado**
¿Considera usted que desarrollar correctamente los casos de uso arquitectónicamente significativos, analizar los casos de uso que tengan fuertes vínculos con la seguridad y realizar una correcta descripción de casos de uso asociados a la seguridad, son los elementos primordiales en el aseguramiento de la seguridad durante la etapa de Análisis y Diseño?					

6. Exprese su criterio acerca de los elementos definidos en la etapa de Implementación.

Criterio de Especialistas					
Pregunta No. 4	**C1 Muy Adecuado**	**C2 Bastante Adecuado**	**C3 Adecuado**	**C4 Poco Adecuado**	**C5 No Adecuado**
¿Considera usted que el acceso a datos, la adecuada selección de las herramientas verticales como los frameworks y asegurar un entorno seguro de desarrollo en el que se va a desarrollar el software, son los elementos críticos que pueden influir en la seguridad del sistema?					

7. Exprese su criterio acerca de los elementos definidos en la etapa de Prueba.

Criterio de Especialistas					
Pregunta No.5	**C1 Muy Adecuado**	**C2 Bastante Adecuado**	**C3 Adecuado**	**C4 Poco Adecuado**	**C5 No Adecuado**
¿Considera usted que la realización de las diferentes pruebas de seguridad como: pruebas de comprobación del sistema de autenticación, pruebas de gestión de sesiones, pruebas de validación de datos, pruebas de gestión de configuración de la infraestructura, pruebas de autorización, pruebas de stress y de carga, pueda influir en la seguridad del sistema?					

8. Exprese su criterio acerca de los elementos definidos en la etapa de Despliegue.

Criterio de Especialistas					
Pregunta No.6	**C1 Muy Adecuado**	**C2 Bastante Adecuado**	**C3 Adecuado**	**C4 Poco Adecuado**	**C5 No Adecuado**
¿Considera usted que la correcta definición de los escenarios de despliegue durante la etapa de Despliegue es un elemento crítico para asegurar la seguridad?					

9. Exprese su criterio acerca de los elementos definidos en la etapa de Soporte.

Criterio de Especialistas					
Pregunta No. 7	**C1 Muy Adecuado**	**C2 Bastante adecuado**	**C3 Adecuado**	**C4 Poco Adecuado**	**C5 No Adecuado**
¿Considera usted que la cantidad de modificacio-nes al software en la etapa de Soporte influya en la seguridad del sistema?					

GLOSARIO DE TÉRMINOS

UCI: Universidad de las Ciencias Informáticas**.**

CEGEL: Centro de Gobierno Electrónico.

TIC*:* Tecnologías de la Información y las Comunicaciones*.*

Experto: Persona, un grupo de ellas u organizaciones capaces de ofrecer valoraciones conclusivas de un área de conocimiento en cuestión y hacer recomendaciones respecto a sus aspectos fundamentales con un máximo de competencia.

ISO: Organización Internacional para la Normalización.

CMMI: Modelo Integrado de Capacidad y Madurez.

Cliente: Una persona u organización, interna o externa a la organización productora que toma responsabilidad financiera por el sistema.

Producto de Software: (IEEE-12207) Es el conjunto de programas de computadora, procedimientos, documentación y datos asociados.

Proceso: (ISO-15504) Proceso de Software, es el proceso (o procesos), usado por una organización (o proyecto), para planificar, administrar, ejecutar, monitorear, controlar y mejorar sus actividades, relacionadas con el software.

Frameworks: Estructuras conceptuales y tecnológicas de soporte definido, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.